CHAPTER

# 2

# IMPLEMENTING, CONFIGURING, AND TROUBLESHOOTING NETWORKING PROTOCOLS

**After reading this chapter and completing the exercises, you will be able to:**

- Describe the history of the TCP/IP protocol stack
- Identify TCP/IP addresses, classes, and subnet masks
- Create a subnetting scheme for a given TCP/IP network address
- Define and assign static and dynamic TCP/IP addresses to Windows 2000 machines
- Establish TCP/IP packet filtering
- Troubleshoot TCP/IP using command-line tools
- Describe the IPX/SPX protocol stack and frame types
- Install and configure NWLink IPX/SPX
- Optimize networking protocol bindings

Before you can begin installing any networking services, you must first install, configure, and sometimes troubleshoot a **protocol stack**. In this chapter, you explore the two most common protocol stacks: TCP/IP and IPX/SPX (implemented in Windows 2000 as NWLink IPX/SPX). In particular, you learn about TCP/IP addresses, from the basics of addressing up to the creation of complex subnetting scenarios. You also discover the NWLink IPX/SPX protocol stack. This chapter introduces you to installation, configuration, and troubleshooting tasks for both of these protocol stacks. Finally, you learn how to optimize networking protocol bindings properly.

## Transmission Control Protocol/Internet Protocol

TCP/IP runs the majority of all major networks today. The ubiquitous nature of the Internet in everyday life brought TCP/IP to the forefront of the information systems world. Created before the **OSI model**, the TCP/IP protocol stack does not follow the seven-layer model. TCP/IP does use a four-layer model that relates to the OSI model. Figure 2-1 shows how the four layers of TCP/IP map to the OSI model.

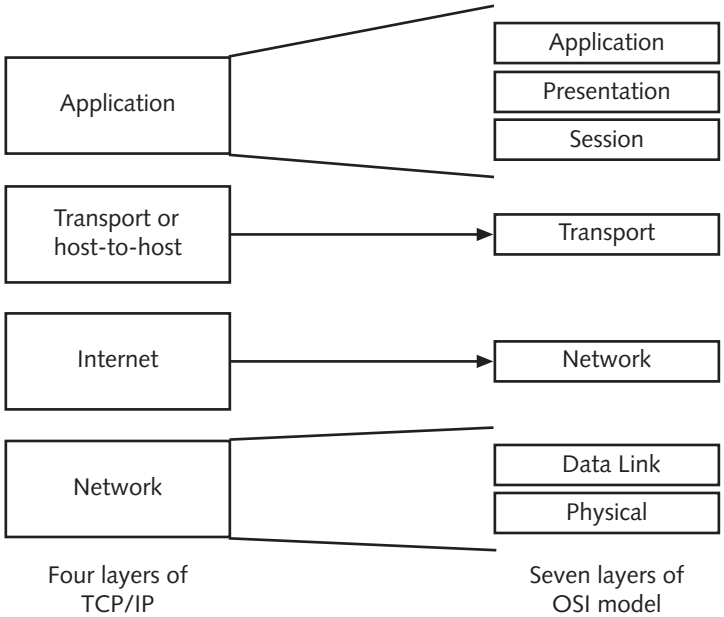



**Figure 2-1** TCP/IP versus OSI model

The Application layer of the TCP/IP model corresponds roughly to the first three layers of the OSI model or its Application, Presentation, and Session layers. The Transport layer or Host-to-Host layer of the TCP/IP model maps one-to-one with the Transport layer of the OSI model. The Internet layer maps to layer 3 or the Network layer of the OSI model. Finally, the Network layer of the TCP/IP model maps to the Data Link and Physical layers of the OSI model. You can easily remember the four layers of the TCP/IP model by using the mnemonic **AT IN** or **A TIN**: **A**pplication, **T**ransport (or host-to-host), **I**nternet, **N**etwork.

Figure 2-2 displays the generic architectural model of TCP/IP with some of the standard protocols supported by Windows 2000 listed. Each protocol within the model performs a specific function for the protocol stack.

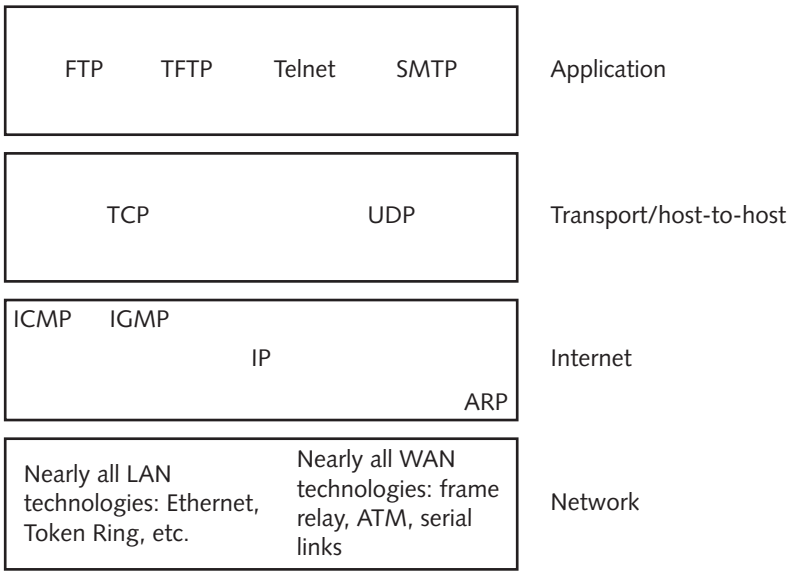



**Figure 2-2** TCP/IP architecture


The model in Figure 2-2 is not Microsoft's exact implementation of TCP/IP. Microsoft does, however, support every protocol shown in the figure.

The following list describes the function of each protocol in Figure 2-2.

- ***File Transfer Protocol (FTP)***: This protocol allows TCP/IP hosts to exchange files between one another. FTP uses TCP as its transport protocol and is therefore connection-oriented and reliable.
- ***Trivial File Transfer Protocol (TFTP)***: Like FTP, TFTP allows hosts to exchange files between one another. However, TFTP uses UDP as its transport protocol and is therefore connectionless and unreliable (unless Application layer processes guarantee delivery).
- ***Telnet***: This application allows a host to log into a remote system and run applications and processes on the remote system.
- ***Simple Mail Transfer Protocol (SMTP)***: The basis for all Internet mail, SMTP provides mail delivery services for the TCP/IP protocol stack.
- ***Transmission Control Protocol (TCP)***: A connection-oriented, reliable transport protocol. TCP sacrifices speed to ensure reliable, error-free transmission of data.
- ***User Datagram Protocol (UDP)***: This connectionless, unreliable transport protocol stresses speed over reliability.

- ***Internet Control Message Protocol (ICMP)***: Providing messaging and communication for protocols within the TCP/IP stack, ICMP handles many communication error messages. PING uses ICMP echo requests and ICMP echo replies to verify that a host is functioning on a TCP/IP network.
- ***Internet Group Management Protocol (IGMP)***: TCP/IP uses IGMP to provide functionality for multicasting. **Multicasting** allows broadcasting of information to specific computers within a multicasting group. IGMP defines group memberships and provides that information to routers.
- ***Internet Protocol (IP)***: This connectionless, layer three protocol determines proper routing within multiple networks.
- ***Address Resolution Protocol (ARP)***: ARP maps a known IP address to a Media Access Control (MAC) layer address.
- ***Local area network (LAN) and Wide area network (WAN) technologies***: TCP/IP does not officially define network layer (OSI Physical layer) technologies. Because of this open approach, nearly all LAN and WAN technologies work with TCP/IP.
- **Connection-oriented protocols**: These protocols guarantee that packets arrive intact, in sequence, and without errors. Connection-oriented protocols sacrifice speed for reliability.
- **Connectionless protocols**: These protocols send packets without regard for guaranteed delivery. A connectionless protocol sacrifices reliability for speed.

## TCP/IP Addressing

TCP/IP addresses, also simply referred to as **IP addresses**, consist of 32 bits normally expressed either as four binary octets separated by periods or as four sets of decimal numbers separated by periods. Therefore each of the following is an example of a valid IP address:

- 192.168.12.8
- 11000000.10101000.00001100.00001000

In fact, each of the examples represents the same IP address. One displays the address as dotted decimal, while the other represents the same address in binary format. You must be able to convert between decimal and binary in order to understand the intricacies of subnetting and IP routing.

The easiest way to convert from binary to decimal or vice-versa involves looking at each octet separately. In each octet, you have the possibility of seven binary bit positions or a maximum of $2^7$. The following example shows the possible values of each bit position within the octet.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |

You can convert the binary IP address 11010000.00110010.11110100.00000111 to decimal by taking each octet and adding together the value of each bit turned on (set to 1). The following example converts the first octet of the binary address above to decimal.

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 128 + | 64 + | 0 + | 16 + | 0 + | 0 + | 0 + | 0 = 208 |

Using this technique, you can convert each octet of the IP address above to a decimal number. The end result is the dotted decimal IP address 208.50.244.7.

Using a slight variation of the first technique, you can also convert from decimal to binary. If you have the IP address, 192.168.24.5, you convert each octet by subtracting each bit position until you are left with a zero remainder. For each bit position you can subtract, you must turn on that bit position (set it to 1). The correct technique for this conversion is shown next:

192 – 128= 64 means the $2^7$ bit position should be turned on (set to a 1).

64 – 64= 0 means the $2^7$ bit position should be turned on (set to 1).

Since the remainder is 0, all other bit positions should be set to off (set to 0). The result is:

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 = 192 |

To convert a decimal to binary correctly, you must subtract the decimal number from the values of each bit position. When the remainder is zero, the conversion is complete and you have your binary number. The following shows the conversion of the second octet, 168, to binary.

- 168 – 128 = 40 means the $2^7$ bit position should be turned on (set to 1).
- 40 – 64 = ??? because you cannot subtract 64 from 40. Therefore, the $2^6$ bit position should be turned off (set to 0).
- 40 – 32 = 8 means the $2^5$-bit position should be turned on (set to 1).
- 8 – 16 = ??? because you cannot subtract 16 from 8. Therefore, the $2^4$ bit position should be turned off (set to 0).
- 8 – 8 = 0 means the $2^3$ bit position should be turned on (set to 1).

Since the remainder is 0, all other bit positions should be set to off (set to 0). The result is:

| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 = 168 |

Now that you can convert between binary and decimal IP addresses (and vice versa), it is imperative that you learn more about the makeup of an IP address. Every IP address contains a **network ID** and a **host ID**. In other words, a portion of each IP address represents the network a particular host is on and a portion signifies the actual host number. This is most easily represented through the use of classes of IP addresses.

## TCP/IP Classes

IP addresses are broken into five different classes. The first three classes reserve a certain portion of the 32 bits available for the network ID and the host ID. The last two classes are used in special situations only. Table 2-1 shows the classes of addresses and several properties of each.

**Table 2-1** TCP/IP classes

| Class of Address | Decimal Range | Leading Bit Values | Number of Networks | Number of Hosts |
|---|---|---|---|---|
| A | 1–126 | 0 | 126 | 16,777,214 |
| B | 128–191 | 10 | 16,384 | 65,534 |
| C | 192–223 | 110 | 2,097,152 | 254 |
| D | 224–239 | 1110 | N/A | N/A |
| E | 240–254 | 11110 | N/A | N/A |

You can see from Table 2-1 that Class A addresses provide a small number of networks consisting of a very large number of hosts. Class B addresses are more balanced. They can accommodate a fairly large number of networks and hosts on each network. Class C addresses provide the largest number of networks, but each individual network can support only 254 possible hosts. You can use the Leading Bit Values column in Table 2-1 to quickly identify a binary number. If, for instance, you encounter an IP address with the first octet shown as 01100010, you know that this entire address is a Class A address. Likewise, if the first octet of an IP addresses consists of 11011010, you can ascertain that the IP address is a Class C address. You can also easily determine the class of address by converting the binary number to a decimal number and using the ranges found in the Decimal Range column of Table 2-1.

Table 2-2 shows what portions of Class A, B, and C networks represent host bits or network bits. The table does not include Class D addresses because they are used for multicasting, nor does it include Class E addresses, which are not currently used in production networks. They are reserved for experimentation.

**Table 2-2** Network ID and host ID

| Class of Address | Network ID | Host ID |
|---|---|---|
| A | First octet | Final three octets |
| B | First two octets | Final two octets |
| C | First three octets | Final octet |

The network ID and host ID in Table 2-2 assume that no subnetting, borrowing of host bits to increase the number of available network bits, is occurring on a network address. Therefore, using the tables you can tell that the address 34.12.4.76 is a Class A address. Also, again assuming no subnetting is occurring, the network ID is 34.0.0.0 and the host ID is .12.4.76.

Again, you know that the number 198.23.54.213 is a Class C address with 198.23.54.0 as the network ID and .213 as the host ID. In short, Class A addresses, by default, have eight bits for the network ID and 24 bits for the host ID. Class B addresses have 16 bits for the network ID and 16 bits for the host ID. Finally, Class C addresses reserve 24 bits for the network ID and eight bits for the host ID.

## Subnet Masks

Subnet masks are the final general component of IP addressing that you must understand to configure TCP/IP. **Subnet masks** are 32-bit numbers that allow hosts on a network to determine which bits in an IP address are network ID bits and which are host ID bits. Subnet masks place a 1 in bit positions that correspond to network ID bits and a zero in bits that represent host ID bits. This allows the host to perform a logical AND function to determine if a destination IP address is on a local network or a remote network. **ANDing** is the process of combining the binary bits in a function very similar to multiplying. If you AND a 1 and a 1, you are left with a 1. ANDing together any other combination (a 1 and 0, or a 0 and 0) gives you a zero. Hosts use subnet masks and the ANDing process to check if destination IP addresses are local or remote. The easiest way to understand this process is with an example.

Host A has an IP address of 192.168.12.4, and Host B has an IP address of 192.168.12.12. If Host A wants to send data to Host B, Host A must determine if the destination IP address is on the same network. Host A first ANDs its IP address with its configured subnet mask. The process works as follows:

| | |
|---|---|
| 11000000.10101000.00001100.00000100 or 192.168.12.4 | Host A's IP address |
| 11111111.11111111.11111111.00000000 or 255.255.255.0 | Host A's subnet mask |
| 11000000.10101000.00001100.00000000 or 192.168.12.0 | the network ID after ANDing |

Host A then ANDs the destination IP address, 192.168.12.12, with its configured subnet mask. The result is:

| | |
|---|---|
| 11000000.10101000.00001100.00001100 or 192.168.12.12 | Host B's IP address |
| 11111111.11111111.11111111.00000000 or 255.255.255.0 | Host A's subnet mask |
| 11000000.10101000.00001100.00000000 or 192.168.12.0 | the network ID after ANDing |

Since the results of the ANDing procedures are the same, Host A knows that Host B is on the exact same network and can attempt to send data directly to Host B. If the two networks IDs are not the same after ANDing, Host A attempts to send data to Host B through its **default gateway**, the IP address of a router port leading to other networks.

Each class of address has a default subnet mask that corresponds to the number of bits assigned as network bits and the number assigned to host bits. (Again, this assumes no additional subnetting is occurring.) Table 2-3 shows the default subnet masks for Class A, B, and C addresses.

**Table 2-3** Default subnet masks

| Class of Address | Default Subnet Mask in Decimal | Default Subnet Mask in Binary |
|---|---|---|
| A | 255.0.0.0 | 11111111.00000000.00000000.00000000 |
| B | 255.255.0.0 | 11111111.11111111.00000000.00000000 |
| C | 255.255.255.0 | 11111111.11111111.11111111.00000000 |

Not all networks use the default subnet masks. Instead, many administrators borrow bits from the host portion of a Class A, B, or C address to create more bits for the network ID. This is called subnetting. Subnetting forces networks to use subnet masks greater than the defaults in Table 2-3.

## SUBNETTING

Subnetting allows administrators to better utilize IP networks that are either assigned to them from the **Internet Assigned Numbers Authority (IANA)** or that they decide to use from the public address space defined by **Request for Comments** (RFC) 1918.


RFC 1918 sets the following addresses aside for use on internal private networks: 10.0.0.0, 172.16.0.0 through 172.31.0.0, and 192.168.0.0 (or 254 possible Class C networks). Marked for use on private networks, these numbers can never be used on the public Internet. In fact, these numbers cannot be routed through the Internet.

The following sections walk you through the process of working with a Class C subnetting scheme and a Class B subnetting scheme.

### Creating Class C Subnetting Scheme

Basic subnetting is very easy when performed in seven steps. This example uses the Class C address 211.212.10.0. Using the seven steps provided here, you can create a subnetting scheme that allows you to use this address on your network.

## Step 1: Determining Number of Subnets Needed

Determining the number of subnets you need is the very first step in subnetting. The number really depends upon your particular network. In Figure 2-3, the network consists of three routers connected via serial links. Each router also has a single Ethernet network attached.

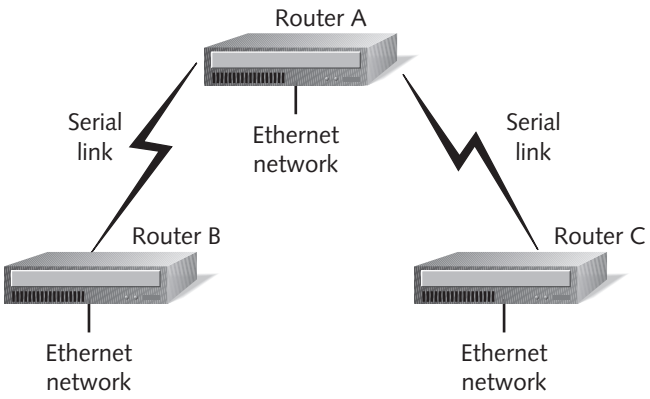



**Figure 2-3** Example of network with three routers

Each shared serial link requires one subnet. Therefore, you need two subnets for the serial links between Router A and Routers B and C. You must also have one subnet per Ethernet interface on each router. Since you have three Ethernet networks, you need three subnets. Using this very simple counting method, you find that you need a total of five subnets. Unfortunately, you have been assigned a Class C address. The network address 211.212.10.0 allows for a single network of 254 hosts. You must borrow host ID bits to make this address work for you.

## Step 2: Determining Number of Bits You Can Borrow

In Step 2, you must determine the number of bits that you can borrow. This number changes depending on the type of network address you start with. For Class A addresses, you have 24 host ID bits, but you can only borrow up to 22. For Class B addresses, you have 16 host ID bits, but, you must leave a minimum of two host bits; therefore, you can borrow 14 bits. Your Class C address (211.212.10.0) has eight total host ID bits, but you can only borrow a maximum of six. The easiest way to determine the number of bits you can borrow is to write the number of octets that contain host ID bits in binary. In the Class C example network 211.212.10.0, you have the following bits to "play" with:

00000000

## Step 3: Determining Number of Bits You Must Borrow to Get Needed Number of Subnets

After you determine the number of subnets you need and the number of bits you can borrow, you must calculate the number of host ID bits you must borrow to get the needed number of

subnets. The formula for determining the number of bits you must borrow is: $2^n-2$ = # of subnets. The *n* represents the number of bits you borrow. In other words, raise two to the power of the number of bits you borrow and subtract two from that number. The result is the number of usable subnets created when you borrow that number of bits.


For nearly all subnetting, it is helpful to place a chart on your page that lists two raised to the powers of 0, 1, 2, up to $2^{13}$. With this chart, you can quickly figure the number of bits you must borrow to get a certain number of subnets.

For the example network, you need five subnets. If you borrow three bits, the formula's result is six usable subnets: $2^3 = 8-2 = 6$.

### Step 4: Turning On Borrowed Bits and Determining Decimal Value

In Step 4, using the bits you determined were available in Step 2, you turn on (set to 1) the number of bits you determined you must borrow in Step 3. You must always begin with the high-order bits (the bits starting on the left of a binary number). Using the number of bits you can work with and the number of bits you must borrow (from Step 3), your result is the following: 11100000. In other words, from the eight total bits from Step 2 (six of which you could borrow), you borrow three host ID bits. In Step 4, you also need to determine the decimal value of the octets from which you borrow host ID bits. In this example, 11100000 equals 224.

### Step 5: Determining New Subnet Mask

Step 5 calculates the new subnet mask after you borrow the host ID bits in Step 4. You must add the decimal value from Step 4 to the default subnet mask for the class of address you are subnetting. This example is a Class C address, so the default mask is 255.255.255.0. The new mask after borrowing three bits becomes 255.255.255.224.

### Step 6: Finding Host/Subnet Variable

In Step 6, you must find the lowest of the high-order bits (bits starting from the left) turned "on." Step 6 takes you all the way back to earlier in the chapter to the values found in each bit position within the octet. Our example defines the octets from which we borrow as 11100000. The highest order bit turned on represents $2^5$, which equals 32. Since $2^5$ is the last high-order bit turned on, the Host/Subnet variable you use in Step 7 is 32.

### Step 7: Determining Range of Addresses

The final step allows you to take the Host/Subnet variable from Step 6 (32) and create your subnet ranges. Using the Class C network above, the range of subnets when you borrow three bits are:

| | | |
|---|---|---|
| 211.212.10.0 | to | 211.212.10.31 |
| 211.212.10.32 | to | 211.212.10.63 |
| 211.212.10.64 | to | 211.212.10.95 |

| | | |
|---|---|---|
| 211.212.10.96 | to | 211.212.10.127 |
| 211.212.10.128 | to | 211.212.10.159 |
| 211.212.10.160 | to | 211.212.10.191 |
| 211.212.10.192 | to | 211.212.10.223 |
| 211.212.10.224 | to | 211.212.10.255 |

IP addresses cannot be all ones or all zeros; therefore, in most cases the first range of addresses and the last range of addresses are unusable. (In some special circumstances, you can use the first range of addresses, or subnet 0. Only certain manufacturers' equipment, such as Cisco Systems, fully supports the use of subnet zero.) In each subnet, the first IP address is unusable because it represents the subnet ID. The final address is also unusable because it is the broadcast address for the subnet. Due to these two restrictions, in subnet one, 211.212.10.33 is the first usable host ID and 211.212.10.62 is the last usable host ID.

## Tailoring a Class B Address

This example takes a Class B address and tries to fit it within the needs of a network containing 1000 subnets. You are assigned the Class B address 131.107.0.0. Using the following seven steps, you are going to subnet the Class B address to meet your needs.

### Step 1: Determining Number of Subnets Needed

Examine your network and determine your needs based on current network configuration and future growth (in this case, 1000 subnets).

### Step 2: Determining Number of Bits You Can Borrow

With this Class B network address, you have 16 total bits to work with. You can only borrow up to 14 of these. On your sheet of paper, you should write the number of bits you have in the host ID portion of the address:

00000000.00000000

### Step 3: Determining Number of Bits You Must Borrow to Get Number of Subnets Needed

Using the formula $2^n-2$ = # of usable subnets, you can easily see that you need to borrow 10 bits. When you plug in 10 borrowed bits, you get the following result:

$2^{10} = 1024 - 2 = 1022$ usable subnets.

### Step 4: Turning on Borrowed Bits and Determining Decimal Value

If you turn on 10 bits, you get the following:

11111111.11000000.

The decimal values for the octets are 255.192.

### Step 5: Determining New Subnet Mask

Your example is a Class B address. In Class B addresses, the default subnet mask is 255.255.0.0. To get your new mask, you add the default mask to the decimal values found in Step 4. The new mask becomes:

255.255.255.192

### Step 6: Finding Host/Subnet Variable

In the next-to-last step, you must find the value of the lowest high-order bit turned on in each octet, from which you borrowed host bits. Since this example is a Class B network and you must borrow a great number of bits to get the proper number of subnets, the borrowing crosses an octet boundary. As a result, you have two Host/Subnet variables. In this example, the variable in the third octet is 1, and the variable for the fourth octet is 64. You get these values by looking at the binary numbers in Step 4. The third octet has the final bit position, or the $2^0$ bit position, turned on. Since $2^0 = 1$, your variable is 1 in the third octet. In the fourth octet, the second high-order bit or $2^6$ is turned on. The variable in this octet is 64.

### Step 7: Determining Range of Addresses

Figuring the range of addresses for Class B networks is much harder than for Class C. This is especially true in cases like this scenario in which you must borrow a large number of bits. Using 1 as the variable in the third octet and 64 as the variable in the fourth octet, the range of the first 9 subnets would be:

| | | |
|---|---|---|
| 131.107.0.0 | to | 131.107.0.63 |
| 131.107.0.64 | to | 131.107.0.127 |
| 131.107.0.128 | to | 131.107.0.191 |
| 131.107.0.192 | to | 131.107.0.255 |
| 131.107.1.0 | to | 131.107.1.63 |
| 131.107.1.64 | to | 131.107.1.127 |
| 131.107.1.128 | to | 131.107.1.191 |
| 131.107.1.192 | to | 131.107.1.255 |
| 131.107.2.0 | to | 131.107.2.63 |

## Static and Dynamic TCP/IP Addresses

You can assign IP addresses to Windows 2000 machines via two main methods: static assignment or dynamic assignment. Each method has certain advantages and disadvantages. In this section, you learn how to use each method to assign an IP address to Windows 2000 machines.

## Static IP Addresses

**Static assignment** is the most work-intensive method for assigning IP addresses, but it also allows the greatest control over address assignment. In order to assign static addresses, you must visit each machine and manually configure the IP address, subnet mask, and, if you have a network with multiple subnets or networks, the default gateway address. Also, in the Windows 2000 environment, you need to configure the DNS settings manually for each machine with a static address.


With static IP addresses, it is imperative that you maintain accurate records on what address has been assigned to each machine. If you do not maintain an accurate database, you can end up having two machines with the same IP address. Since this causes a multitude of communication problems, you must ensure that your static IP list is accurate and up-to-date. Certain machines, such as domain controllers, Web servers, and most application servers, require that the machine have a static IP address. Later in this book, you learn that nearly every networking service requires the server to have a static IP address.

To begin the process of assigning static IP addresses, you need to access the properties for your local area connections. You open the Local Area Connection Status dialog box by clicking Start, Settings, Network and Dial-up Connections, Local Area Connection.


You can also access Local Area Connection and open the Local Area Connection status box by right-clicking the **My Network Places** icon on the desktop and then double-clicking Local Area Connection. Regardless of the method used, you need to click the Properties button in the Local Area Connection Status dialog box to access the network settings.

Figure 2-4 shows the **Local Area Connection Properties** dialog box where you actually configure your TCP/IP settings with a static address.

You must click Internet Protocol (TCP/IP) and click the Properties button to assign a static IP address. Figure 2-5 shows the Internet Protocol (TCP/IP) Properties dialog box.

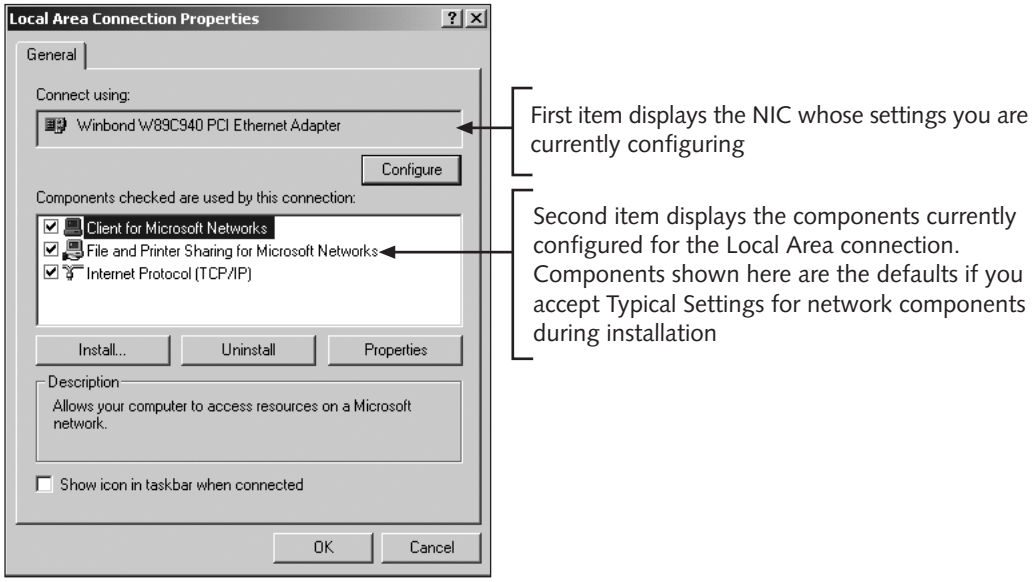



**Figure 2-4** Local Area Connection Properties

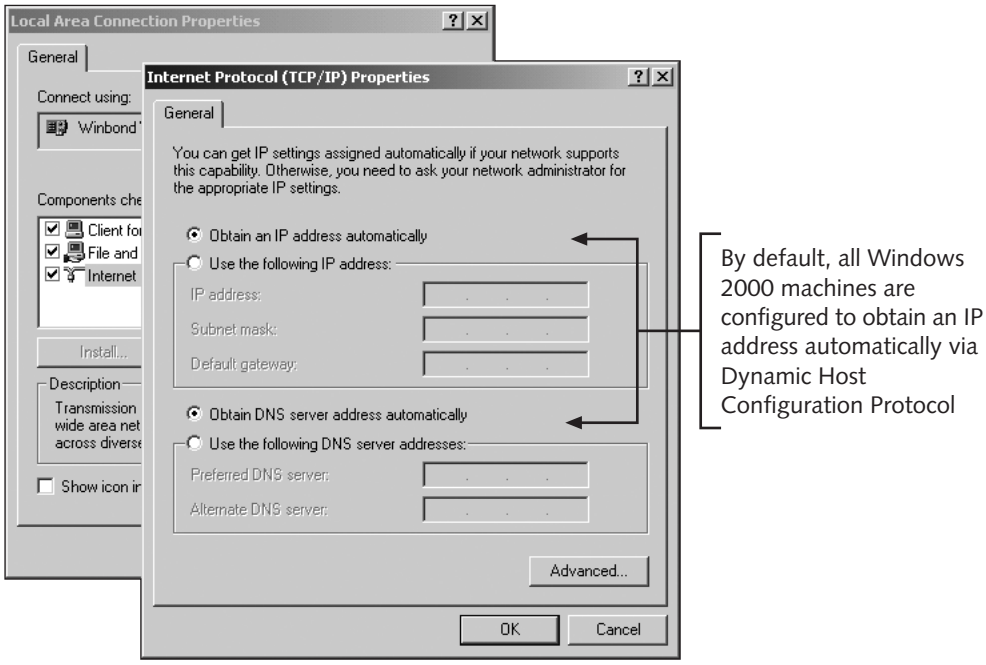



**Figure 2-5** Internet Protocol (TCP/IP) Properties dialog box

Once you obtain a unique static IP address, you can assign it by clicking the Use the following IP address radio button. Figure 2-6 shows the configuration of the static IP address, 192.168.0.200.

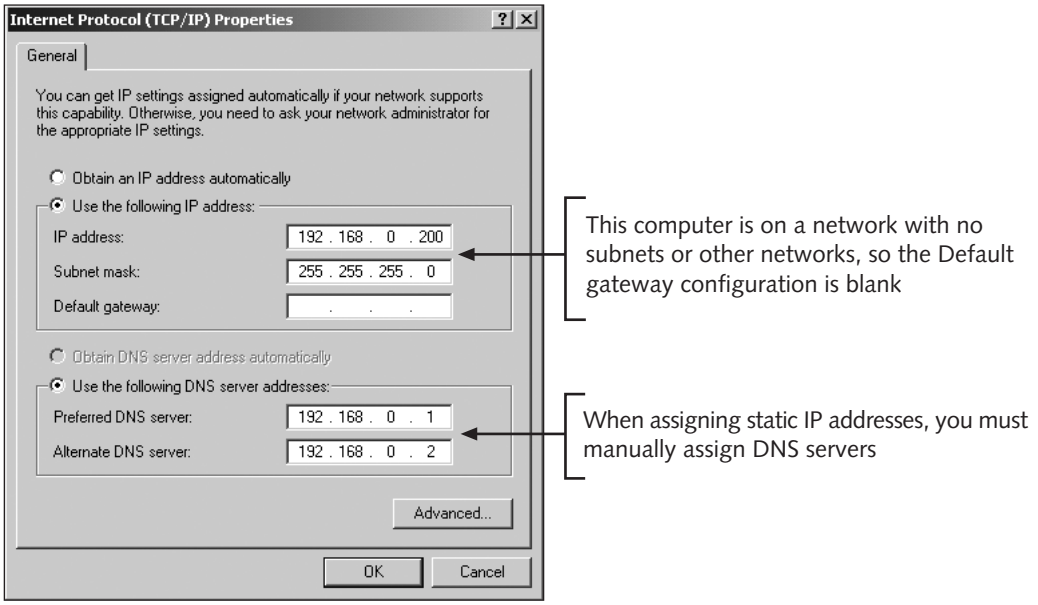



**Figure 2-6** Configuring a static IP address

Once you click **OK**, your address is configured. Unlike Windows NT 4.0, which requires you to restart your machine, your static IP address is immediately configured and usable on the Windows 2000 machine. In Figure 2-6, the Advanced button is in the lower-right corner of the Internet Protocol (TCP/IP) Properties dialog box. If you click the Advanced button, the Advanced TCP/IP settings dialog box opens, as shown in Figure 2-7.

You can assign multiple static IP addresses to a single NIC and assign multiple default gateways in the Advanced TCP/IP Settings dialog box. Windows 2000 supports automatic Dead Gateway Detection: if a Windows 2000 machine determines that its primary default gateway is unreachable, it switches to a secondary gateway if one has been configured.

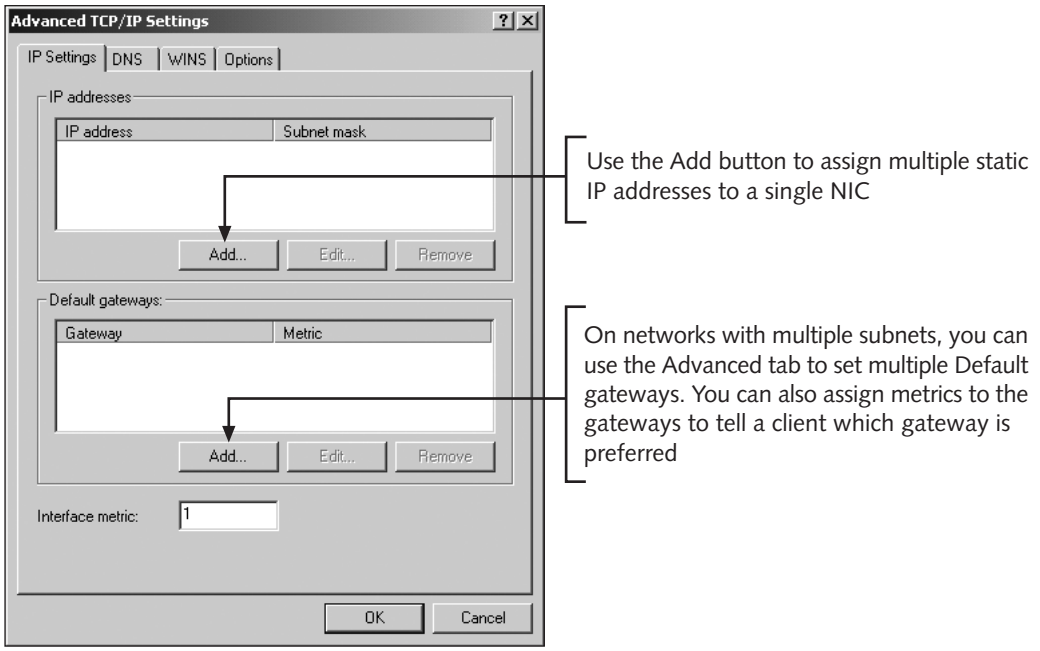



**Figure 2-7** Advanced TCP/IP Settings dialog box

## Dynamic IP Addresses

**Dynamic assignment** is much less labor intensive than static assignment. The fact that you can easily configure clients with IP addresses without visiting each machine is a huge advantage of dynamic IP addresses. Since you can also assign options such as DNS servers, default gateways, and WINS servers via dynamic addressing, you can save an enormous amount of time. The downside of dynamic addresses involves the use of a **Dynamic Host Configuration Protocol** or DHCP server to hand out addresses. Chapter 3 discusses in detail DHCP servers and the actual process a client goes through to obtain a dynamic address. For now, you just need to know that dynamic IP address assignment requires an installed and properly configured DHCP server on your network.

Configuring a Windows 2000 machine for a dynamic IP address is very easy. You must navigate to the Internet Protocols (TCP/IP) Properties dialog box. Once there, you simply click the Obtain an IP address automatically and Obtain DNS server address automatically buttons. (You are not required to obtain DNS information automatically, but most clients using a dynamic IP address also get their DNS information dynamically.) Figure 2-8 shows the correct settings for a client configured to obtain a dynamic IP address.

Although dynamic IP addressing is by far the easiest method of IP address assignment, it is not appropriate for all Windows 2000 machines. As mentioned in the "Static IP Addresses" section, some machines, such as the server running the DHCP server service, must use a static address. Still, to lessen the administrative overhead of IP addressing, you should use dynamic IP addresses on as many machines on your network as possible.

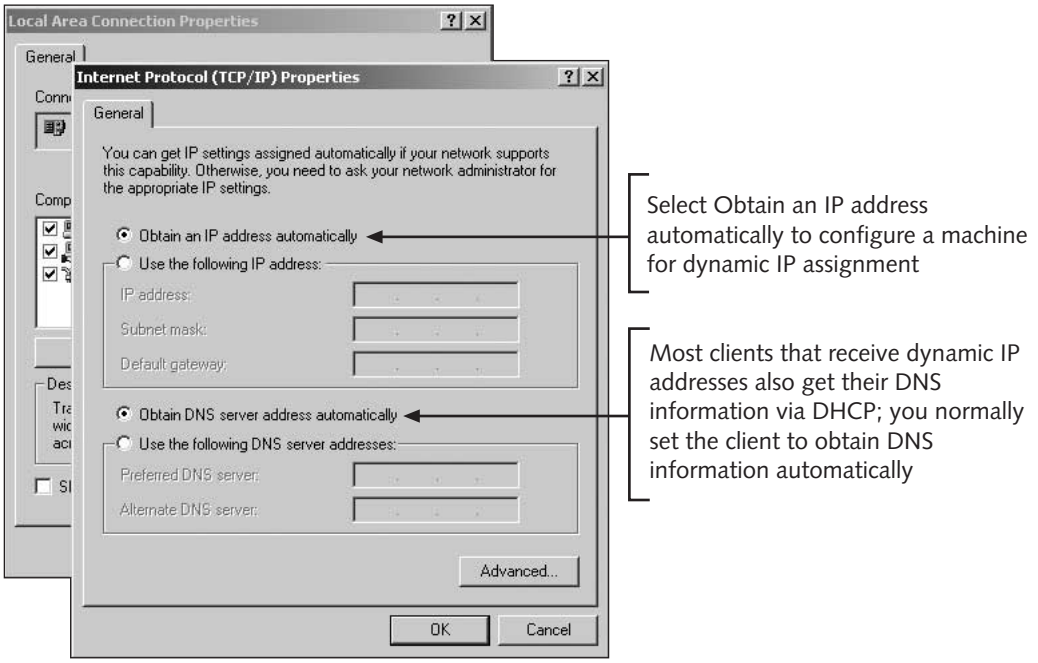



**Figure 2-8** Configuring a dynamic IP address

## TCP/IP Packet Filtering

Windows 2000 provides support for TCP/IP packet filtering. TCP/IP packet filtering allows you to control the types of incoming IP traffic that all network interface cards in a machine will process. TCP/IP in Windows 2000 provides very rudimentary packet filtering that is usually only implemented if no other type of packet filtering is utilized.

You must access the Advanced TCP/IP setting dialog box to enable TCP/IP packet filtering. Then click the Options tab, and double-click TCP/IP filtering to work with the filtering options. Figures 2-9 displays the TCP/IP Filtering dialog box.

Since you can filter by TCP port, UDP port, or IP protocols, you must carefully plan exactly what incoming traffic you wish to filter. Suppose, for example, that you want to block all incoming TCP traffic to a particular machine, except for Web traffic. Allowing TCP port 80, the hypertext transport protocol port, is the easiest way to accomplish this task. Figure 2-10 shows TCP/IP filtering configured to allow only Web traffic.

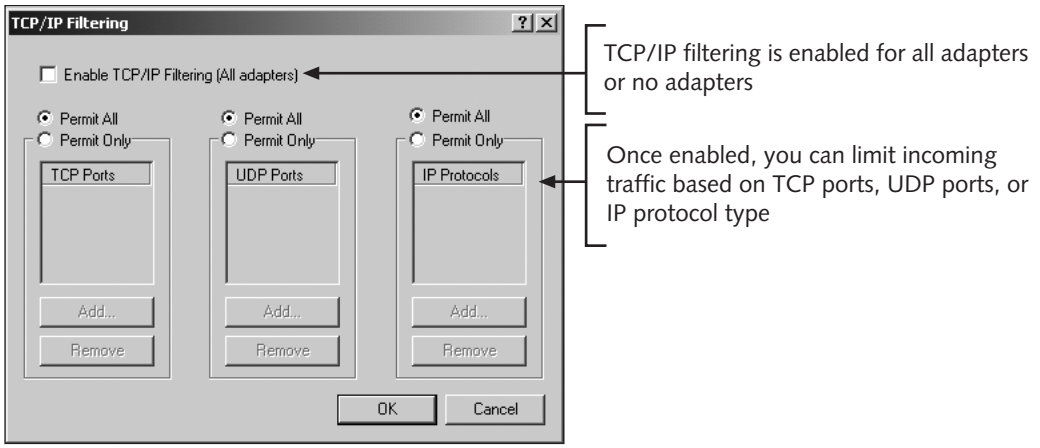



**Figure 2-9** TCP/IP Filtering dialog box

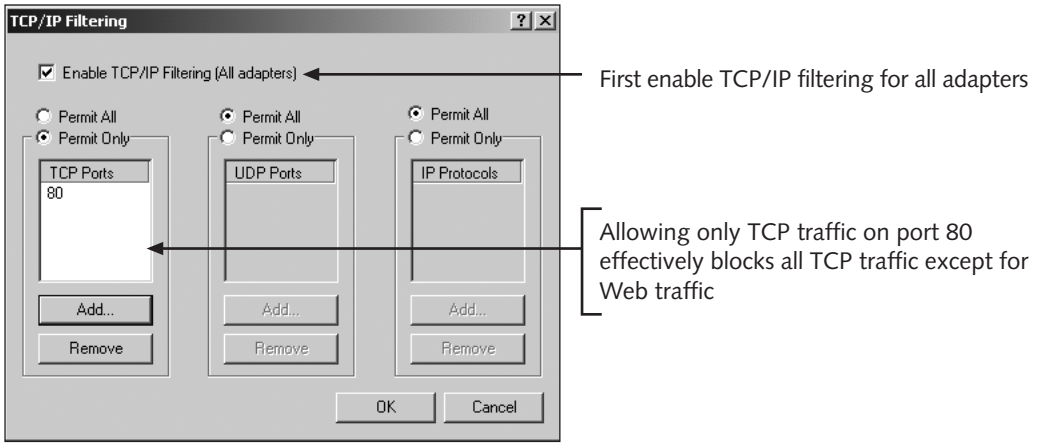



**Figure 2-10** Allowing Web traffic with TCP/IP filtering

Using careful planning, you can allow only incoming traffic that is absolutely necessary for a particular machine. You must use caution, however. Once you decide to permit only certain TCP ports, UDP ports, or IP protocols, all ports or protocols not explicitly defined are not allowed. You can very easily block necessary ports or protocols by mistake.

## TROUBLESHOOTING TCP/IP

Windows 2000 includes a variety of command-line tools that you can use to test and troubleshoot TCP/IP. These tools allow you to do everything from verifying IP configuration on the local machine to testing connectivity with a remote host. This section discusses the command-line tools ipconfig, ping, tracert, netstat, nbtstat, netdiag, and pathping.

Troubleshooting TCP/IP is a five-step process. Other utilities, like nslookup, hostname, and route, are "*real world*" tools that you can use in addition to the five steps. They are discussed later in this book.



In most cases, it takes a combination of command-line tools to test a machine fully. The following five steps are recommended to troubleshoot TCP/IP related problems:

1. Run ipconfig/all.
2. Ping the loopback. This tells you if TCP/IP has been loaded.
3. Ping the local IP address.
4. Ping a host on the same network.
5. Ping a remote host.

If you start all troubleshooting with the ipconfig /all command, you can verify that a machine with a static address has been configured with the correct IP address, subnet mask, and default gateway information. The ipconfig /all command also allows you to determine if clients configured to get dynamic IP addresses have actually received addresses. Step 2, pinging the loopback, is a bit redundant because the ipconfig /all command tells you if TCP/IP has been loaded, but pinging the loopback guarantees that the machine has TCP/IP loaded and initialized. Step 3 helps you determine if you did indeed load the correct address on a statically assigned machine or that you do have a dynamically assigned address. Step 4 tells you that the machine is connected with your local LAN and that the correct network and subnet mask is configured in its IP address. Finally Step 5, ping a remote host, ensures that the default gateway is configured correctly and that routing is active between your network and other networks.

## Ipconfig Commands

One of the first troubleshooting tasks is to verify that the TCP/IP configuration has been entered correctly. This is especially important for machines configured with static IP addresses. Mistyping an address is very easy when you must manually enter each IP address. Ipconfig is a command-line tool that, among other things, displays the current IP configuration on a Windows 2000 machine. All command-line utilities run from the **command prompt**, a command-line interface to Windows 2000. You can find the command prompt by clicking Start, Programs, Accessories, Command Prompt. Once you open the command prompt, you can issue the ipconfig /? command to see all the switches available. You get the following output from the ipconfig /? command:

```
Windows 2000 IP Configuration

USAGE:
   ipconfig [/? | /all | /release [adapter] | /renew
[adapter]
            | /flushdns | /registerdns
```

```
                | /showclassid adapter
                | /setclassid adapter [classidtoset] ]

   adapter    Full name or pattern with '*' and '?' to 'match',
              * matches any character, ? matches one
character.
   Options
       /?                  Display this help message.
       /all                Display full configuration
                           information.
       /release            Release the IP address for the
                           specified adapter.
       /renew              Renew the IP address for the
                           specified adapter.
       /flushdns           Purges the DNS Resolver cache.
       /registerdns        Refreshes all DHCP leases and re-
                           registers DNS names
       /displaydns         Displays the contents of the DNS
                           Resolver Cache.
       /showclassid        Displays all the dhcp class IDs
                           allowed for adapter.
       /setclassid         Modifies the dhcp class id.

The default is to display only the IP address, subnet mask
and default gateway for each adapter bound to TCP/IP.

For Release and Renew, if no adapter name is specified, then
the IP address leases for all adapters bound to TCP/IP will
be released or renewed.

For SetClassID, if no class id is specified, then the
classid is removed.

Examples:
    > ipconfig                  ... Show information.
    > ipconfig /all             ... Show detailed information
    > ipconfig /renew           ... renew all adapters
    > ipconfig /renew EL*       ... renew adapters named EL..
..
    > ipconfig /release *ELINK?21* ... release all matching
adapters,e.g.ELINK-21, myELELINKi21adapter.
```

This output shows the wide variety of switches available to change the functionality of the ipconfig command. The most used switches are /all, /release, /renew, and /registerdns.

## ipconfig /all

If you start all troubleshooting with the ipconfig /all command, you can verify that machines with static addresses are configured with the correct IP address, subnet mask, and default gateway information. The ipconfig /all command also allows you to determine if clients configured to get dynamic IP addresses actually received addresses. The correct command syntax and output received from the ipconfig /all command are:

```
H:\>ipconfig /all

Windows 2000 IP Configuration

        Host Name . . . . . . . . . . . . : win2kpro
        Primary DNS Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Hybrid
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
Description . . . . . . . . . . . : Winbond W89C940 PCI
Ethernet Adapter

        Physical Address. . . . . . . . . : 00-20-78-11-4A-62
        DHCP Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        IP Address. . . . . . . . . . . . : 192.168.0.26
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.0.1
        DHCP Server . . . . . . . . . . . : 192.168.0.1
        DNS Servers . . . . . . . . . . . : 192.168.0.1
        Primary WINS Server . . . . . . . : 192.168.0.1
        Lease Obtained. . . . . . . . . . : Saturday, May 27,
        2000 1:58:21 PM
       Lease Expires . . . . . . . . . . : Tuesday, May 30,
       2000 1:58:21 PM
```

This command provides a large amount of information about the IP configuration of the machine. In the lower portion of the output, you can see the Media Access Control address of the machine, IP address, subnet mask, default gateway, DHCP server, DNS server, and WINS server. Using this information, you can see that this particular machine is configured for dynamic addresses via DHCP. Once you know that, you can run the ipconfig /release command to release the currently configured IP address.

## ipconfig /release

The ipconfig /release command produces the following output:

```
H:\>ipconfig /release

Windows 2000 IP Configuration

IP address successfully released for adapter "Local Area
Connection"
```

At this point you can run the ipconfig /all command to verify that the IP address has actually been released. The following results show that the machine is indeed no longer configured with a dynamically assigned IP address.

```
        Windows 2000 IP Configuration

              Host Name . . . . . . . . . . . . : win2kpro
              Primary DNS Suffix  . . . . . . . :
              Node Type . . . . . . . . . . . . : Hybrid
              IP Routing Enabled. . . . . . . . : No
              WINS Proxy Enabled. . . . . . . . : No

        Ethernet adapter Local Area Connection:
              Connection-specific DNS Suffix  . :
              Description . . . . . . . . . . . : Winbond W89C9
              40 PCI Ethernet Adapter
              Physical Address. . . . . . . . . : 00-20-78-11-
              4A-62
DHCP Enabled. . . . . . . . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IP Address. . . . . . . . . . . . : 0.0.0.0
Subnet Mask . . . . . . . . . . . : 0.0.0.0
Default Gateway . . . . . . . . . :
DHCP Server . . . . . . . . . . . : 255.255.255.255
DNS Servers . . . . . . . . . . . :
            Primary WINS Server . . . . . . . : 192.168.0.1
```

## ipconfig /renew

To obtain a dynamically assigned IP address after you release all addresses, run the ipconfig /renew command. This command forces the machine to acquire a new IP address. (Of course, if the old address is still available, the client may end up configured with the old address.) Acquisition of DHCP addresses is discussed in more detail later.

## ipconfig /registerdns

New to the ipconfig command is the /registerdns switch. This switch provides the functionality described in the /release and /renew switches, but it also refreshes dynamic DNS name registrations. Since Windows 2000 uses DDNS as its primary naming tool, this command is very useful.


If you wish to save a copy of the information provided by the output of any command discussed in this section, you can use the > character to redirect output from the command to a text file. For instance, if you want the information provided by ipconfig in a text file, you type ipconfig /all >d:\ipconfig.txt. The item after the > is the complete path to the file you wish to contain the information.

## ping

The **Packet Internet Groper**, or ping, command is the second important troubleshooting command you need to understand. The ping command verifies connectivity with remote hosts. It does this via ICMP echo requests and ICMP echo replies. Whenever you issue the ping command, you send a series of four ICMP echo requests to the designated host. If connectivity is possible, the host returns an ICMP echo reply for each request. The following command output displays the ping command and the responses you get from a successful reply.

```
C:\>ping 192.168.12.2

Pinging 192.168.12.2 with 32 bytes of data:

Reply from 192.168.12.2: bytes=32 time<10ms TTL=128
Reply from 192.168.12.2: bytes=32 time<10ms TTL=128
Reply from 192.168.12.2: bytes=32 time<10ms TTL=128
Reply from 192.168.12.2: bytes=32 time<10ms TTL=128

Ping statistics for 192.168.12.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>
```

If you issue the ping command and the ICMP echo replies are not returned, you get the following output:

```
C:\>ping 192.168.12.1

Pinging 192.168.12.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.12.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms

C:\>
```

If the ping command fails, it is important to begin troubleshooting both the basic network configuration of the host (and remote host) and the IP configuration. Begin with simple troubleshooting. Check to see if the host has a network cable patched into a network jack. Also use the ipconfig command to verify IP settings.

The ping command, like the ipconfig command, has a large number of command line switches. When you type ping /?, you receive the following list of switches:

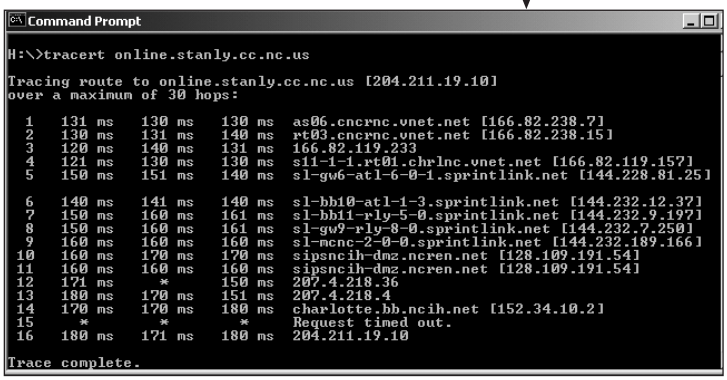
```
                H:\>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL]
[-v TOS]
            [-r count] [-s count] [[-j host-list] |
[-k host-list]]
            [-w timeout] destination-list

Options:
     -t             Ping the specified host until stopped.
                    To see statistics and continue -
                    type Control-Bre
                    To stop - type Control-C.
     -a             Resolve addresses to hostnames.
     -n count       Number of echo requests to send.
     -l size        Send buffer size.
     -f             Set Don't Fragment flag in packet.
     -i TTL         Time To Live.
     -v TOS         Type Of Service.
     -r count       Record route for count hops.
     -s count       Timestamp for count hops.
     -j host-list   Loose source route along host-list.
     -k host-list   Strict source route along host-list.
     -w timeout     Timeout in milliseconds to wait for each
                    reply.
```

You can use these switches in a variety of ways. The ping –t command starts a continuous string of ICMP echo requests. This can be useful if you wish to monitor connectivity for a short time period. The –l switch allows you to specify the size of the ICMP packet sent. Each switch follows the syntax ping –[*switch*] *destination IP address*. Ping should be the first command you use to verify connectivity with a remote host.

## tracert

The tracert (or trace route) command traces the route a packet takes to a remote system with the use of ICMP echo requests and an incremental Time To Live (TTL) counter. Tracert first sends an ICMP echo request packet with a TTL of 1. Once this packet crosses the first router, it times out and returns. Tracert then sends an ICMP packet with a TTL of 2, 3, and so on. Tracert, by default, allows a maximum TTL of 30. (Using the –h # *of hops* command line switch, you can increase the number of hops to above 30). If a destination host is more than

30 routers or "hops" away, it is considered unreachable. The syntax for the tracert command is simply tracert *w.x.y.z*, where *w.x.y.z* is the IP address of the destination host, the host whose path you wish to learn. Figure 2-11 shows the output from the tracert command and the path to a particular host.




The tracert command displays the path to a particular host plus information on round trip times. The round trip time information can be used to find out if a slow link is causing communication problems. Also, the tracert may reveal that a path to the remote host stops after a particular hop. In this example, if the host had not been found after the series of time-outs (signified by three*) on hop 15, the administrators would know to contact the administrator for the router found in hop 14 to see if there are any known problems

**Figure 2-11** Output from tracert command

Tracert allows local administrators to determine if problems with communication are on the local area network, the wide area network, or the Internet service provider's network. If the tracert command returns successful values for all paths on the internal LAN but has time-outs on the way to a particular host, administrators know that the problems are not on their networks. Instead, in this case, congestion on wide area links provided by ISPs is probably causing the problem.

## netstat

Netstat displays information about a host's established TCP/IP connections. The following output shows a TCP/IP host that used a Web browser to access Course Technology's Web site at *www.course.com*.

```
H:\>netstat

Active Connections
Proto  Local Address   Foreign Address        State
TCP    win2kpro:1113   199.95.72.8:http       ESTABLISHED
TCP    win2kpro:1114   199.95.72.8:http       ESTABLISHED
TCP    win2kpro:1044   HOMEPDC:netbios-ssn    ESTABLISHED
```

The workstation, win2kpro, accessed the Web site at 199.95.72.8 via TCP using http. In other words, the user of this station is browsing Course Technology's Web page. Netstat also has a series of switches that you can use to customize it. You can view the switches using the netstat /? switch.

If you only want to see TCP connections to a machine, you could issue the netstat –p tcp command. Likewise, you can view only UDP connections with the netstat –p udp command. Netstat allows you to quickly view what resources are either accessing a workstation or being accessed by a workstation.

## nbtstat

Since NetBIOS naming is still an integral part of Windows 2000 networks, Microsoft continues to provide the nbtstat command-line tool for viewing NetBIOS over TCP/IP connection information.

Nbtstat allows you to view the currently open NetBIOS connections on a machine. You must run the nbtstat command with either the –a switch, which requires you to specify the NetBIOS name of the machine you want NetBIOS information on, or with the –A switch, which lets you specify the IP address of the machine you want information about. Output from the nbtstat command is similar to the following output.

```
H:\>nbtstat -a win2kpro

Local Area Connection:
Node IpAddress: [192.168.0.26] Scope Id: []

        NetBIOS Remote Machine Name Table
   Name               Type               Status
---------------------------------------------------
   WIN2KPRO           <00>  UNIQUE       Registered
   WORKGROUP          <00>  GROUP        Registered
   WORKGROUP          <1E>  GROUP        Registered
   WIN2KPRO           <20>  UNIQUE       Registered
   WORKGROUP          <1D>  UNIQUE       Registered
   ..__MSBROWSE__.    <01>  GROUP        Registered
   WIN2KPRO           <03>  UNIQUE       Registered
   ADMINISTRATOR      <03>  UNIQUE       Registered

   MAC Address = 00-20-78-11-4A-62
```

Using the NetBIOS codes and the information the nbtstat command (along with many of its switches presented), you can troubleshoot NetBIOS naming problems. The WINS section of this book discusses in greater detail the nbtstat command as well as NetBIOS naming.

## netdiag


Netdiag is only available on a Windows 2000 machine if the Windows 2000 Support Tools are loaded from the \support\tools folder on the Windows 2000 CD-ROM.

A new troubleshooting command in Windows 2000 is netdiag. This command performs a series of tests on the networking components of a system. The tests check many parts of the networking configuration including such items as all configured protocols. In fact, netdiag tests items such as NDIS, WINS, DNS, trusts, modems, and even the IPX/SPX protocol stack. Although switches and information on them are available using the netdiag /? command, one of the best ways to run the command is with the netdiag >*d*:*filename.txt*. (*d* is the name of the local drive where you want to save the information.) Output from this command follows.

```
                ..............................

        Computer Name: WIN2KPRO
        DNS Host Name: win2kpro
        System info : Windows 2000 Professional (Build 2195)
        Processor : x86 Family 6 Model 5 Stepping 2, GenuineIntel
        List of installed hotfixes :
            Q147222

    Netcard queries test . . . . . . . : Passed
    [WARNING] The net card 'RAS Async Adapter' may not be work-
    ing because it has not received any packets.


    Per interface results:

        Adapter : Local Area Connection

            Netcard queries test . . . : Passed

            Host Name. . . . . . . . . : win2kpro
            IP Address . . . . . . . . : 192.168.0.26
            Subnet Mask. . . . . . . . : 255.255.255.0
            Default Gateway. . . . . . : 192.168.0.1
            Primary WINS Server. . . . : 192.168.0.1
            Dns Servers. . . . . . . . : 192.168.0.1

            AutoConfiguration results. . . . . . : Passed
```

```
            Default gateway test . . . : Passed

            NetBT name test. . . . . : Passed
                No remote names have been found.

            WINS service test. . . . . : Passed

        Adapter : {4DDF24E5-6F69-4B69-95AB-ABACD6BD9D8E}

            Netcard queries test . . . : Passed

            Host Name. . . . . : win2kpro
            IP Address . . . . . : 166.82.50.90
            Subnet Mask. . . . . : 255.255.255.255
            Default Gateway. . . . . : 166.82.50.90
            NetBIOS over Tcpip . . . . : Disabled
            Dns Servers. . . . . : 166.82.1.3
                                   166.82.1.8


            AutoConfiguration results. . . . . : Passed

            Default gateway test . . . : Passed

            NetBT name test. . . . . : Skipped
                NetBT is disabled on this interface. [Test
skipped]

            WINS service test. . . . . : Skipped
                NetBT is disable on this interface. [Test
skipped].


    Global results:


    Domain membership test . . . . . : Passed
        Dns domain name is not specified.
        Dns forest name is not specified.

    NetBT transports test. . . . . : Passed
        List of NetBt transports currently configured:
            NetBT_Tcpip_{85507B45-378F-45FA-BF7C-58C97784ED5A}
        1 NetBt transport currently configured.


    Autonet address test . . . . . : Passed
```

```
IP loopback ping test. . . . . . . : Passed

Default gateway test . . . . . . . : Passed

NetBT name test. . . . . . . . . . : Passed

Winsock test . . . . . . . . . . . : Passed

DNS test . . . . . . . . . . . . . : Passed

Redir and Browser test . . . . . . : Passed
    List of NetBt transports currently bound to the Redir
        NetBT_Tcpip_{85507B45-378F-45FA-BF7C-58C97784ED5A}
    The redir is bound to 1 NetBt transport.

    List of NetBt transports currently bound to the browser
        NetBT_Tcpip_{85507B45-378F-45FA-BF7C-58C97784ED5A}
    The browser is bound to 1 NetBt transport.

DC discovery test. . . . . . . . . : Skipped

DC list test . . . . . . . . . . . : Skipped

Trust relationship test. . . . . . : Skipped

Kerberos test. . . . . . . . . . . : Skipped

LDAP test. . . . . . . . . . . . . : Skipped

Bindings test. . . . . . . . . . . : Passed

WAN configuration test . . . . . . : Passed
Entry Name: Vnet
Device Type: Framing protocol :  PPP
LCP Extensions :  Disabled
Software Compression :  Enabled
```

```
Network protocols :
     NetBEUI
     IPX
     TCP/IP
IP Address :  Specified
Name Server: Specified
IP Header compression :  Enabled
Use default gateway on remote network : Enabled

     Connection Statistics:
     Bytes Transmitted     : 497371
     Bytes Received        : 1124089
     Frames Transmitted    : 4956
     Frames Received       : 4486
     CRC    Errors         : 4486
     Timeout Errors        : 0
     Alignment Errors      : 1
     H/W Overrun Errors    : 0
     Framing Errors        : 0
     Buffer Overrun Errors : 0
     Compression Ratio In  : 2
     Compression Ratio Out : 5
     Baud Rate ( Bps )     : 31200
     Connection Duration   : 4779162


Modem diagnostics test . . . . . : Passed

IP Security test . . . . . : Passed
    IPSec policy service is active, but no policy is assigned.


The command completed successfully
```

Netdiag actually performs many of the same functions already offered by other troubleshooting tools such as nbtstat, netstat, and the ipconfig /all command. Its /fix switch can also solve some trivial DNS problems. In fact, many switches associated with netdiag prove invaluable troubleshooting tools. The following output lists all netdiag switches.

```
H:\>netdiag /?

Usage: netdiag [/Options]>
   /q - Quiet output (errors only)
   /v - Verbose output
   /l - Log output to NetDiag.log
   /debug - Even more verbose.
   /d:<DomainName> - Find a DC in the specified domain.
   /fix - fix trivial problems.
```

```
   /DcAccountEnum - Enumerate DC machine accounts.
   /test:<test name>  - tests only this test. Non -
skippable tests will still be run
   Valid tests are :-
        Ndis - Netcard queries Test
        IpConfig - IP config Test
        Member - Domain membership Test
        NetBTTransports - NetBT transports Test
        Autonet - Autonet address Test
        IpLoopBk - IP loopback ping Test
        DefGw - Default gateway Test
        NbtNm - NetBT name Test
        WINS - WINS service Test
        Winsock - Winsock Test
        DNS - DNS Test
        Browser - Redir and Browser Test
        DsGetDc - DC discovery Test
        DcList - DC list Test
        Trust - Trust relationship Test
        Kerberos - Kerberos Test
        Ldap - LDAP Test
        Route - Routing table Test
        Netstat - Netstat information Test
        Bindings - Bindings Test
        WAN - WAN configuration Test
        Modem - Modem diagnostics Test
        Netware - Netware Test
        IPX - IPX Test
        IPSec - IP Security Test
   /skip:<TestName> - skip the named test.  Valid tests are:
        IpConfig - IP config Test
        Autonet - Autonet address Test
        IpLoopBk - IP loopback ping Test
        DefGw - Default gateway Test
        NbtNm - NetBT name Test
        WINS - WINS service Test
        Winsock - Winsock Test
        DNS - DNS Test
        Browser - Redir and Browser Test
        DsGetDc - DC discovery Test
        DcList - DC list Test
        Trust - Trust relationship Test
        Kerberos - Kerberos Test
        Ldap - LDAP Test
        Route - Routing table Test
        Netstat - Netstat information Test
        Bindings - Bindings Test
        WAN - WAN configuration Test
        Modem - Modem diagnostics Test
```

```
Netware - Netware Test
IPX - IPX Test
IPSec - IP Security Test
```

The netdiag /v /l command provides you, as an administrator, with a huge amount of information concerning network configuration and status. This command creates verbose output and saves it in a file called NetDiag.log in the root directory of the active drive (the drive on which the system files reside).

### pathping

The last command-line tool discussed in this section is a new tool called pathping. This command combines functions of the ping command and the tracert command. You can display its command syntax with the pathping /? command. When you run the command, its output is similar to the following.

```
H:\>pathping 192.168.0.1

Tracing route to HOMEPDC [192.168.0.1]
over a maximum of 30 hops:
  0  win2kpro [192.168.0.26]
  1  HOMEPDC [192.168.0.1]

Computing statistics for 25 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0
win2kpro [192.168.0.26]
       0/ 100 =  0%
           1    0ms     0/ 100 =  0%        0/ 100 =  0%
HOMEPDC [192.168.0.1]

Trace complete.
```

The | item in statistics displays the packet loss during the round trip from source to destination. In this example (pathping between two computers on the same network), both the round trip time (RTT) and packet loss rate are extremely good. Over the Internet or on a busy LAN, RTT and packet loss may be considerably worse. The pathping command allows you to run the same test available with the ping or trace command, but with only a single command.

## Internetwork Packet eXchange/Sequenced Packet Exchange

On many networks today, TCP/IP is not the only protocol stack running. In fact, most networks consist of heterogeneous components from many different manufacturers. One of the most common network configurations today is one with Windows NT 4.0 servers and Novell Netware servers running concurrently. Another common configuration is Netware

servers and Windows NT 4.0 workstation clients. The people at Microsoft knew that these configurations would continue to be used, so they built into Windows 2000 the same support available now for the Netware/NT/Windows 2000 hybrid networks. *The MCSE Guide to Microsoft Windows 2000 Server*, ISBN: 0-619-01517-9 from Course Technology discusses in detail such items as Client Services for Netware and Gateway Services for Netware. This book focuses on what IPX/SPX is and how Windows 2000 implements it as NWLink IPX/SPX.

Novell developed IPX/SPX from a protocol stack created by Xerox. Novell needed a protocol stack to run its newly developed Netware network operating system. IPX/SPX uses an 80-bit address format consisting of a network.node format. The first 32 bits form the network portion of the address. The last 48 bits are the node (or host) portion of the address. In fact, the node portion of the address is made up of the media access control address of the node. The following is an example of an IPX/SPX address:

200.0020.7811.4a62

In the address, 200 is the network id, while the last 12 hexadecimal digits (48 bits) represent the node or MAC address of the client. Note that the network address does not occupy the entire 32-bit range. While the network id can be up to 32 bits, it does not need to be a full 32 bits in length.

When you begin to configure NWLink IPX/SPX (the Microsoft 32-bit implementation of IPX/SPX), you need to know the frame type used by other IPX/SPX clients on your network. IPX/SPX and NWLink IPX/SPX support the four frame types listed in Table 2-4.

**Table 2-4** Novell supported frame types

| Novell IPX/SPX Frame Types | Used by: |
|---|---|
| 802.3 | Networks running NetWare 3.11 or lower |
| 802.2 | Networks running NetWare 3.12 or higher |
| Ethernet_II | Networks running both IPX/SPX and TCP/IP |
| Ethernet_snap | Networks running IPX/SPX, TCP/IP, and AppleTalk |

Later, in the section, "Installing and Configuring NWLink IPX/SPX," you learn that you can set a Windows 2000 machine to use either Auto Frame Type detection or you can manually set the frame type. A common problem with Windows 2000 machines occurs when a network running IPX/SPX uses multiple frame types. If you set a Windows 2000 machine to Auto Frame Type detection on a network with multiple frame types, it only configures and uses the 802.2 frame type. This causes the machine to be unable to "see" any machines running other frame types. In this scenario, you should manually configure each frame type.

The architecture of IPX/SPX, like the TCP/IP protocol stack, does not follow the OSI model exactly. Instead, as Figure 2-12 shows, it maps very loosely to the OSI model.

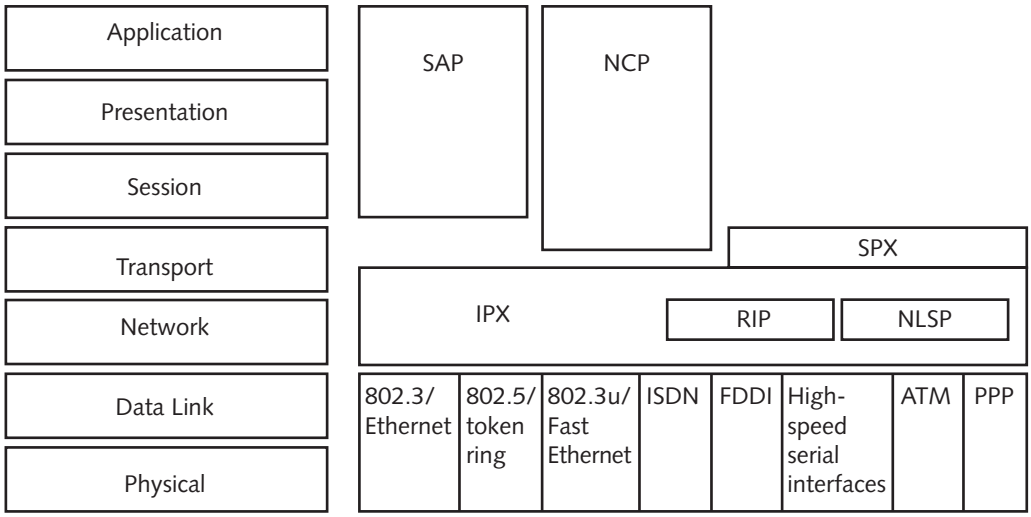



**Figure 2-12** IPX/SPX protocol stack architecture

The following list describes the main protocols of the IPX/SPX protocol stack.

- ***Internetwork Packet eXchange (IPX)***: IPX is a connectionless, predominately layer 3 protocol, although as you can see in Figure 2-12, it does assume some layer 4 functions. It is responsible for finding the best path through a multipath IPX network. IPX is similar in function to the IP protocol found in the TCP/IP protocol stack. It can use RIP and NLSP to determine the best path among multiple paths through the internetwork.
- ***Sequenced Packet eXchange (SPX)***: SPX, a connection-oriented, layer 4 protocol, provides guaranteed delivery services for the connectionless IPX protocol. SPX is similar in function to TCP in the TCP/IP protocol stack.
- ***Service Advertisement Protocol (SAP)***: SAP, an upper-layer protocol (layers 5, 6, and 7), advertises services running on IPX/SPX servers and helps clients locate network services.
- ***NetWare Core Protocol (NCP)***: NCP facilitates client/server interaction on a NetWare network. NCP handles basic file and print sharing, authentication services, and directory services. NCP functions at layers 4, 5, 6, and 7 of the OSI model.
- ***Routing Information Protocol (RIP)***: RIP is an integrated, distance-vector, routing protocol that uses ticks (1/18 of a second time counts) and hop count as metrics to determine the best path within an IPX/SPX internetwork. In IPX/SPX, RIP sends routing table updates every 60 seconds. RIP functions at layer 3 of the OSI model.

- ***NetWare Link State Protocol (NLSP)***: Novell designed NLSP, a link state routing protocol, as the successor to RIP. Like RIP, NLSP functions at layer 3 of the OSI model.

Now that you have a basic understanding of the IPX/SPX protocol stack, you must learn how to configure NWLink IPX/SPX on Windows 2000 machines. The next section covers installing and configuring this protocol.

## Installing and Configuring NWLink IPX/SPX

If you choose the typical network settings during installation, Windows 2000 only installs the Client for Microsoft Networks, File and Printer Sharing for Microsoft Networks, and TCP/IP. To install other protocol stacks such as NWLink IPX/SPX, you must access the Local Area Connection Properties. First access the Local Area Connection icon by right-clicking My Network Places and clicking Properties. Then, right-click the Local Area Connection icon and click Properties to access the Install icon. Figure 2-13 shows how to access the dialog box for the Local Area Connection properties.

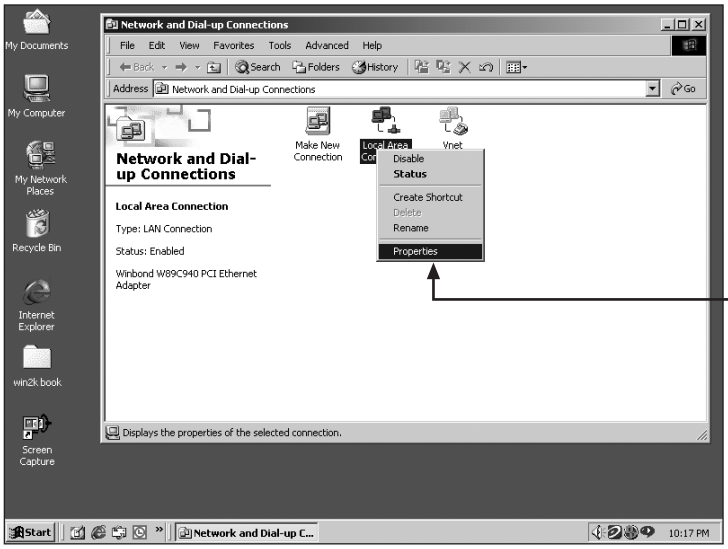



**Figure 2-13** Accessing Local Area Connection properties

After clicking Properties, you see the Local Area Connection Properties dialog box shown in Figure 2-14.

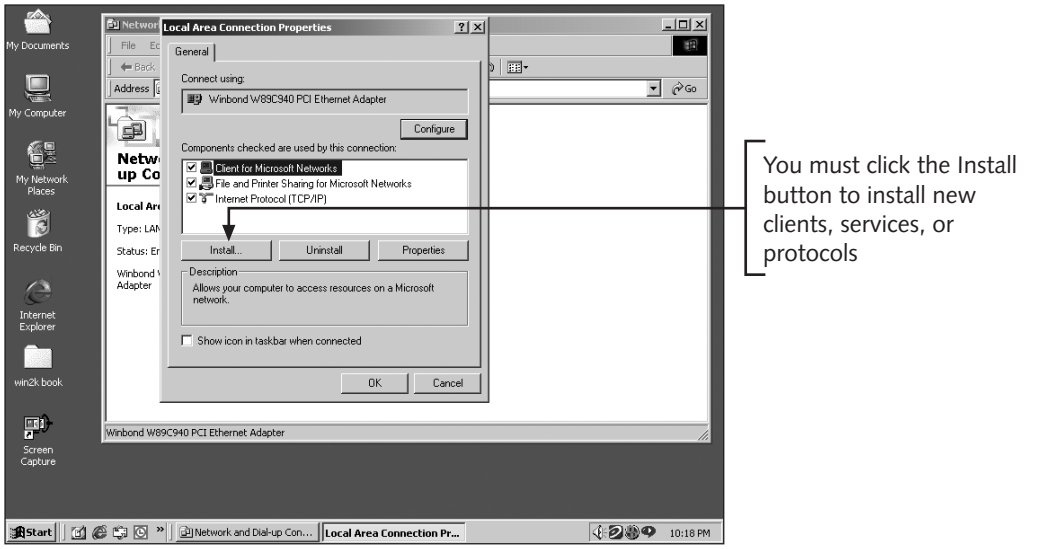



**Figure 2-14** Local Area Connection Properties dialog box

To install the NWLink IPX/SPX (or any other client, service, or protocol, for that matter), you must click Install to access the Select Network Component Type dialog box. You can either select Protocol and click the Add button, or double-click Protocol to open the Select Network Protocol dialog box shown in Figure 2-15.

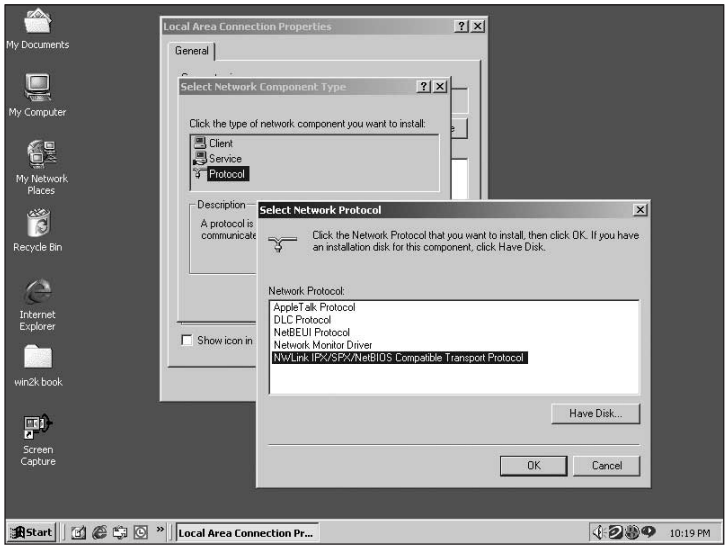
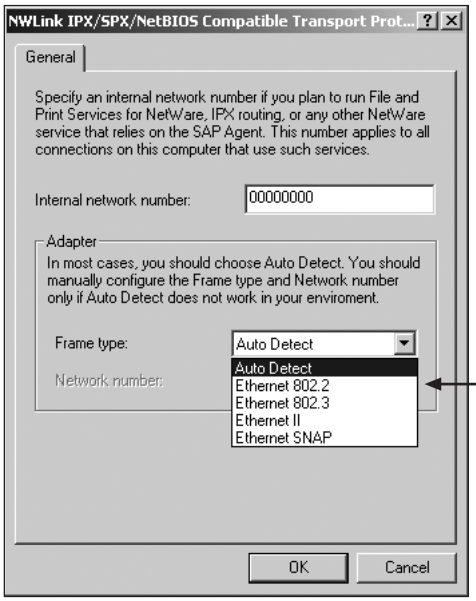



**Figure 2-15** Select Network Protocol dialog box

At this point, NWLink IPX/SPX is installed with Frame type set to Auto Detect. If your environment requires the use of multiple frame types, you must manually assign the frame types and network number. You accomplish this task by accessing the properties for the NWLink IPX/SPX/NetBIOS Compatible Transport Protocol Configuration dialog box. Figure 2-16 shows this box.




On a network with multiple IPX/SPX frame types, you must select each frame type manually and may also need to manually configure the Network number for each frame type. The Novell Netware administrator should be able to provide this information for you

**Figure 2-16** NWLink IPX/SPX/NetBIOS Compatible Transport Protocol Configuration dialog box

NWLink IPX/SPX is now installed and configured on your Windows 2000 machine, but if you need to connect to Netware servers, you still need to install either one or both of the additional services provided for Netware: Client Services for Netware or Gateway Services for Netware.

## Network Protocol Bindings

**Binding** is the process of associating or connecting a particular protocol or service to a network adapter card. Each networking protocol on a Windows 2000 machine must be bound to at least one NIC. The rules for optimizing protocol bindings are very simple. First, you should move your most used protocols up in the protocol binding order. To do this, you must access the Advanced Settings from the Advanced command in Network and Dial-up Connections, as shown in Figure 2-17.

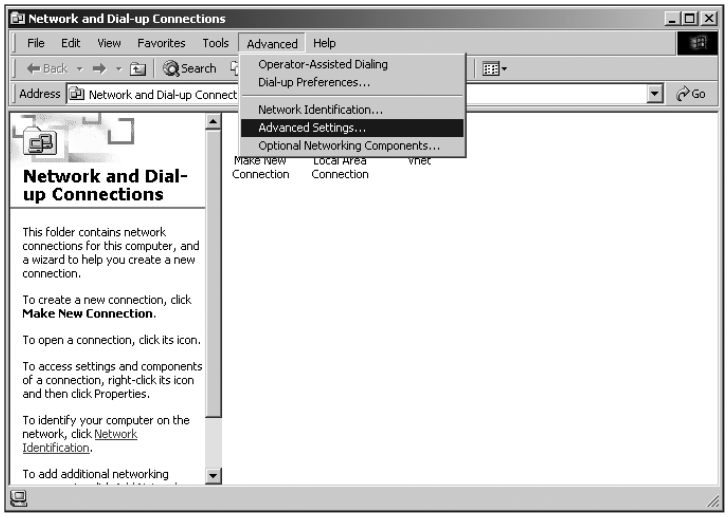



**Figure 2-17** Using Advanced Settings for network and dial-up connections

Once you click the Advanced Settings, you open the Advanced Settings dialog box shown in Figure 2-18.

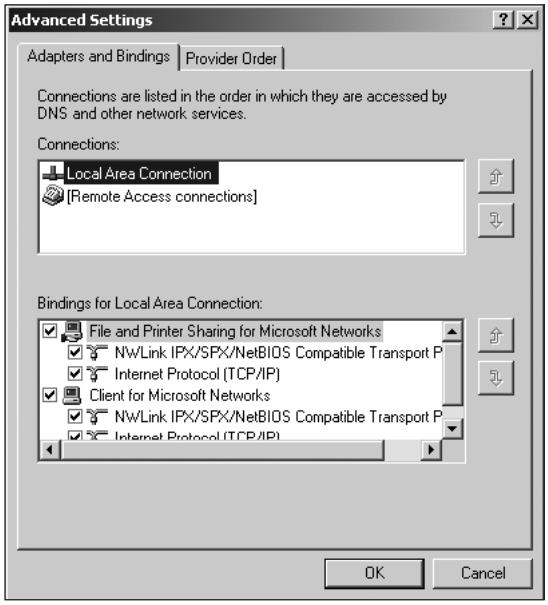



**Figure 2-18** Advanced Settings dialog box

In Figure 2-18 NWLink IPX/SPX is above the Internet Protocol (TCP/IP) in the binding order. If NWLink IPX/SPX is the protocol predominantly used by all servers and clients on

the network, then the binding order is set correctly. On most Windows 2000 networks, the Internet Protocol (TCP/IP) is the most used protocol. To correct the binding order, you click Internet Protocol (TCP/IP) and click the right most up arrow to move TCP/IP up in the binding order. Figure 2-19 shows the same Advanced Settings dialog box after you move TCP/IP up in the binding order.

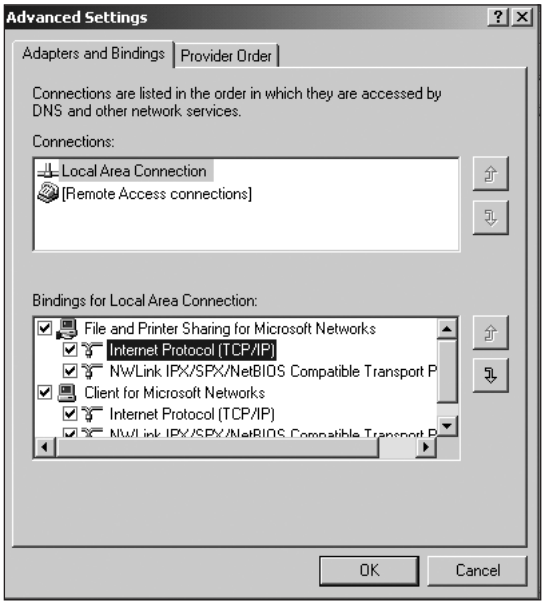



**Figure 2-19** Advanced Settings dialog box after moving up Internet Protocol (TCP/IP) in the binding order

Removing unused bindings is the second rule for optimizing network binding. If you are not going to use a protocol for a particular service, you should remove it by deselecting it in the Adapters and Bindings area of the Advanced Settings dialog box. Refer to Figures 2-17 through 2-19 and assume that you have decided that you do not need to use NWLink IPX/SPX for either File and Printer Sharing for Microsoft Networks or the Client for Microsoft Networks components. You can disable NWLink IPX/SPX by removing the check mark from the check box next to the protocol's name.

## Chapter Summary

- Implementing, configuring, and troubleshooting networking protocols is an essential task in preparing a Windows 2000 Professional or Windows 2000 server machine for participation in a network. Windows 2000 supports a multitude of networking protocols, but two of the most important are TCP/IP and IPX/SPX.

- TCP/IP is a robust, scalable, mature internetworking protocol developed originally by the ARPANet project. It is designed to allow dissimilar systems to "talk" to one another. IP addresses are the 32-bit logical addresses that, along with subnet masks, must be installed on all clients on a TCP/IP network to allow communication to occur. These addresses can be set either as static addresses or dynamic addresses. The choice between the two depends mainly on the desired function of the Windows 2000 machine. (Most clients use dynamic; most servers use static.) During IP address assignment, you can set optional components such as DNS servers and default gateways. In addition, using advanced features such as TCP/IP packet filtering, network administrators can control what types of incoming packets access various machines.
- Windows 2000 includes numerous command-line tools to help troubleshoot TCP/IP once it is installed. Ipconfig can verify IP settings while ping can test connectivity with remote hosts. Tracert, or trace route, tests and displays the path between two TCP/IP hosts on the same network or different networks. Netstat displays a list of current TCP/IP connections. Nbtstat can check NetBIOS over TCP/IP configurations and settings. Netdiag, a new command, gives administrators access to a wealth of information about the current network configuration of a machine. Pathping combines the best of tracert with ping, providing detailed statistics about the connection between two hosts.
- TCP/IP is not the only protocol supported by Windows 2000. NWLink IPX/SPX allows Windows 2000 machines to communicate on networks running the IPX/SPX protocol stack. NWLink IPX/SPX supports all four major frame types found in IPX/SPX: 802.3, 802.2, Ethernet_II, and Ethernet_SNAP. Installing NWLink is easy, but you must be careful when setting the frame type to Auto. On networks with multiple IPX/SPX frame types configured, this causes communication problems. In these cases, you must set the frame type and network number manually.
- Finally, administrators must carefully manage protocol bindings to ensure optimal performance by network components. The most used protocols should be moved up in the binding order and any unnecessary binding should be removed.

## Key Terms

**Address Resolution Protocol (ARP)** — Lower-layer protocol that resolves a known IP address to a MAC address.

**Advanced Research Projects Agency Network (ARPANet)** — Original name for the Internet; ARPA was the government agency responsible for sponsoring the research that lead to the TCP/IP protocol stack and the modern-day Internet.

**ANDing** — Logically combining binary numbers; the results are similar to multiplying binary numbers; ANDing a 1 and a 1 gives a 1. All other combinations (1 and 0, and 0 and 0) result in 0.

**binary format** — IP address displayed as four sets of eight binary numbers separated by periods.

**binding** — Process of associating a protocol with a NIC or a network service.

**dead gateway detection** — Feature of Windows 2000 that allows a machine to detect when a default gateway is unreachable and then switch to a configured back-up default gateway.

**default gateway** — IP address of the router port to networks outside the local network.

**dotted decimal** — IP addresses displayed as a series of four decimal numbers separated by periods, for example, 192.168.12.2.

**dynamic assignment** — Configuring a host to obtain an IP address automatically using DHCP.

**Dynamic Host Configuration Protocol (DHCP)** — Protocol used by clients to obtain IP addresses dynamically from a DHCP server.

**Ethernet** — Most widely used networking architecture; contention-based architecture that uses carrier sense multiple access/collision detection as its access method.

**File Transfer Protocol (FTP)** — Provides for file transfer between two TCP/IP hosts; uses TCP as its transport protocol.

**host ID** — Portion of an IP address that represents the bits used for host identification.

**Internet Assigned Numbers Authority (IANA)** — Group responsible for controlling allocation of IP addresses to the Internet community.

**Internet Control Message Protocol (ICMP)** — Handles the communication of errors and status messages within the TCP/IP protocol stack.

**Internet Group Management Protocol (IGMP)** — TCP/IP protocol used to establish and maintain multicasting groups.

**Internet Protocol (IP)** — Connectionless, best-effort delivery protocol in the TCP/IP protocol stack that handles routing of data and logical addressing with IP addresses.

**Internetwork Packet eXchange (IPX)** — Connectionless, layer three protocol that provides routing function for the IPX/SPX protocol stack.

**IP address** — 32-bit logical addresses that must be assigned to every host on a TCP/IP network.

**ipconfig** — Command-line tool used to verify IP settings; can also be used to renew or release dynamically assigned IP addresses and DNS information.

**local area network (LAN)** — Network confined within a small area such as a single building or a small campus.

**Media Access Control (MAC) address** — Physical address burned into the EPROM on a network interface card.

**multicasting** — Broadcasting packets to only certain hosts on a TCP/IP network.

**nbtstat** — Command-line tool that displays NetBIOS over TCP/IP information.

**netdiag** — New command-line tool in Windows 2000 that tests a large portion of the networking components on a machine. Provides much of the same information as other command-line tools such as netstat, nbtstat, and ipconfig.

**netstat** — Command-line tool that provides information about current TCP/IP connections.

**Netware Core Protocol (NCP)** — Primary upper-layer protocol in IPX/SPX that facilitates client/server interaction.

**Netware Link State Protocol (NLSP)** — More advanced link state routing protocol in the IPX/SPX protocol stack Designed to replace the RIP protocol.

**network ID** — Portion of an IP address that represents the bits reserved for the network number.

**OSI model** — Open Systems Interconnection model, a theoretical model for the process two machines go through when communicating with one another over a network.

**Packet Internet Groper (ping)** — Command-line tool used to test connectivity between two IP hosts.

**pathping** — Command-line tool that combines ping and tracert functions with new statistics reporting functions.

**protocol stack** — Group of protocols working together to complete the network communication process.

**Request for Comments (RFC)** — Proposals presented to the Internet community describing everything from possible TCP/IP standards to simple informative tracts.

**Routing Information Protocol (RIP)** — Routing protocol provided with the IPX/SPX protocol stack.

**Sequenced Packet eXchange (SPX)** — Layer four protocol that provides guaranteed delivery; similar in function to TCP.

**serial links** — Generally slow-speed connections used for wide area network connectivity.

**Service Advertisement Protocol (SAP)** — Protocol used on IPX/SPX networks by clients to find network services and by servers to advertise network services.

**Simple Mail Transfer Protocol (SMTP)** — Application layer TCP/IP protocol that provides mail delivery services.

**static assignment** — Manually assigning an IP address to a host.

**subnet mask** — 32-bit number used to determine the portion of an IP address that represents the network ID and the host ID.

**subnetting** — The process of borrowing host bits to increase the number of network bits.

**telnet** — Application layer protocol in TCP/IP that allows a user to log on to a remote host and execute programs remotely.

**tracert** — Trace route command-line tool that allows testing of the entire path between two hosts.

**Transmission Control Protocol (TCP)** — Transport layer protocol in the TCP/IP protocol stack that is connection-oriented and reliable; provides guaranteed delivery.

**Trivial File Transfer Protocol (TFTP)** — Like FTP, provides file transfer between two TCP/IP hosts; TFTP uses UDP as its transport protocol and is faster, but more unreliable than FTP.

**User Datagram Protocol (UDP)** — Connectionless, best-effort delivery transport layer protocol in the TCP/IP stack.

**wide area network (WAN)** — Network or collection of networks spread across a large geographical area.

## Review Questions

1. Which one of the following command-line tools can you use to release and renew dynamically assigned IP addresses?
   a. ping
   b. ipconfig
   c. netdiag
   d. pathping
2. What class of address is the IP address 135.12.5.4?
   a. A
   b. B
   c. C
   d. None of the above
3. Which of the following does assigning a static IP address require? (Choose all that apply.)
   a. A properly configured DHCP server
   b. Manually visiting the machine to be configured
   c. An IP address, subnet mask, and optional default gateway
   d. The ipconfig /release command
4. What is the default subnet mask for a Class B network?
   a. 255.255.255.0
   b. 0.0.0.0
   c. 255.255.0.0
   d. 255.0.0.0
5. Which of the following are possible frame types on an IPX/SPX network? (Choose all that apply.)
   a. NCP
   b. 802.2
   c. Ethernet_II
   d. 802.3
6. Which TCP/IP protocol provides connection-oriented, guaranteed delivery, transport layer services?
   a. TCP
   b. UDP
   c. IP
   d. FTP

7. Which of the following decimal numbers represents the binary number 11011001?
   a. 213
   b. 217
   c. 205
   d. None of the above
8. Which command-line tool displays information concerning nearly every networking component on a system?
   a. ipconfig
   b. netstat
   c. nbtstat
   d. netdiag
9. The four layers of the TCP/IP protocol stack model are _____, _____, _____, and _____.
10. You can use the ping command to verify network connectivity between two TCP/IP hosts. True or false?
11. Which of the following should be done to optimize network bindings? (Choose all that apply.)
    a. Move most used protocols up in the binding order.
    b. Add as many protocols as possible, more is better.
    c. Remove or deselect protocols from any unnecessary networking component.
    d. Do nothing, Windows 2000 auto configures protocol binding with optimal settings.
12. If you borrow 10 bits from the host portion of the IP network 10.0.0.0, what is the new default subnet mask?
    a. 255.0.0.0
    b. 255.255.255.0
    c. 255.255.224.0
    d. 255.255.192.0
13. You added NWLink IPX/SPX to your system, but it cannot communicate with some IPX/SPX hosts on your network. Other IPX/SPX clients can see the hosts in question. Which one of the following is the most likely cause of the problem on your Windows 2000 machine?
    a. The binding for IPX/SPX has been removed.
    b. NWLink IPX/SPX is set to Auto Frame type detection.
    c. TCP/IP is interfering with the NWLink IPX/SPX protocol stack.
    d. The hosts on the network are configured incorrectly.

14. You need three subnets and you have the private address space 192.168.12.0. What is the range of the first usable subnet?
    a. 192.168.12.0 to 192.168.12.31
    b. 192.168.12.10 to 192.168.12.15
    c. 192.168.12.32 to 192.168.12.63
    d. 192.168.12.64 to 192.168.12.95
15. What are advantages of dynamically assigned IP addresses? (Choose all that apply.)
    a. Excessive administrative overhead
    b. Easy configuration of options
    c. Need for a DHCP server
    d. Easy client configuration
16. Which one of the following shows the correct allocation of network ID bits and host ID bits in a Class A network?
    a. Network.network.host.host
    b. Network.host.host.host
    c. Network.network.network.host
    d. None of the above
17. Which of the following must be configured on machines with static addresses if they are on a network that is not connected to any other subnets or networks? (Choose all that apply.)
    a. Subnet mask
    b. Unique IP address
    c. Default gateway
    d. DNS server information
18. Which protocol communicates error and informational messages within the TCP/IP protocol stack?
    a. IGMP
    b. TCP
    c. ARP
    d. ICMP
19. The first octet of an IP address is 01110110. What class of address is this IP address?
20. List the three private address spaces provided for in RFC 1918.

## HANDS-ON PROJECTS

All Hands-on Projects in this chapter require two computers set up as described in the lab set-up section in the front of this book. For these exercises, you use the PCs named win2kpro1 and win2kdc02.


### Project 2-1

**To install a static address on the Windows 2000 machine named win2kdc02:**

1. Right-click **My Network Places** and choose **Properties**, as shown in Figure 2-20.

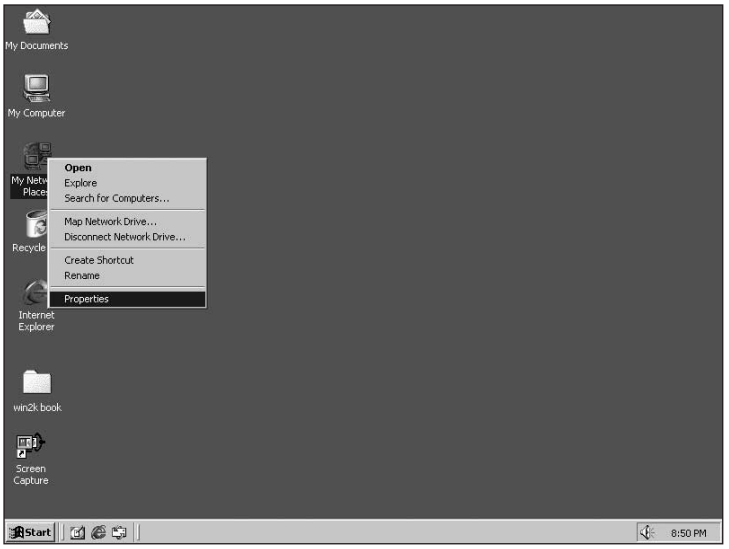



**Figure 2-20** Accessing network and dial-up properties via My Network Places

2. Right-click **Local Area Connection** and choose **Properties**.
3. Click **Internet Protocol (TCP/IP)** and then choose **Properties**.
4. In the General section of the Internet Protocol (TCP/IP) Properties dialog box (shown in Figure 2-21), select the **Use the following IP Address**.
5. In the IP address: field, enter **192.168.12.2**, and press **Tab**.

   Windows 2000 automatically fills the subnet mask with 255.255.255.0.
6. Click **OK** to close the Internet Protocol (TCP/IP) Properties dialog box.
7. Click **OK** to close the Local Area Connection Properties box.

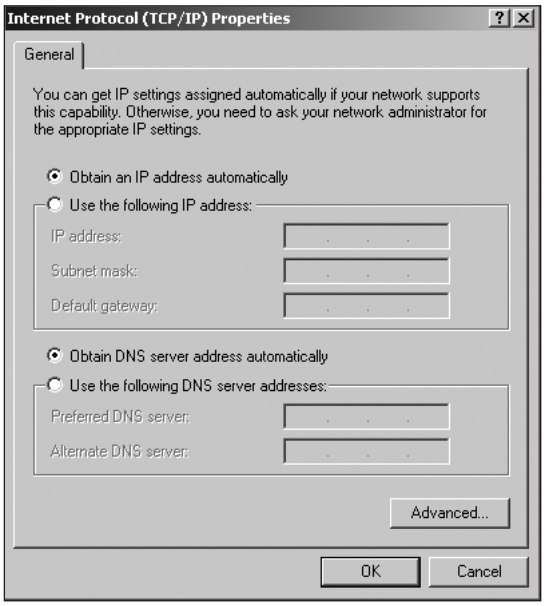



**Figure 2-21** Internet Protocol (TCP/IP) Properties dialog box

8. Close the Networking and Dialup Connections dialog box by clicking the **Close** button in its upper-right corner.

Repeat the steps in Hands-on Project 2-1 on the machine named win2kpro1. In Step 5, use the IP address 192.168.12.10.


## Project 2-2

**To verify that the static address in Project 2-1 is configured correctly:**

1. Click **Start** and choose **Programs**, **Accessories**, **Command Prompt**.
   The command prompt in Figure 2-22 should appear.


**Figure 2-22** Command prompt

2. After the Command Prompt, type **ipconfig /all** and press **Enter**.
3. Verify on win2kdc02 that the IP address is **192.168.12.2** and the subnet mask is **255.255.255.0**.

Repeat the steps in Hands-on Project 2-2 on the machine named win2kpro1. In Step 3 verify that the IP address is 192.168.12.10 and the subnet mask is 255.255.255.0.


## Project 2-3

This Hands-on Project requires completion of Hands-on Projects 2-1 and 2-2.

**To test connectivity between hosts win2kdc02 and win2kpro:**

1. From win2kdc01, click **Start** and choose **Programs**, **Accessories**, **Command Prompt**.
2. After the command prompt, type **ping 192.168.12.10** and press **Enter**.

   If you correctly configured the IP addresses in Hands-on Project 2-1 and verified them in Hands-on Project 2-2, you should receive successful replies to the ping command. Figure 2-23 displays successful ping replies.

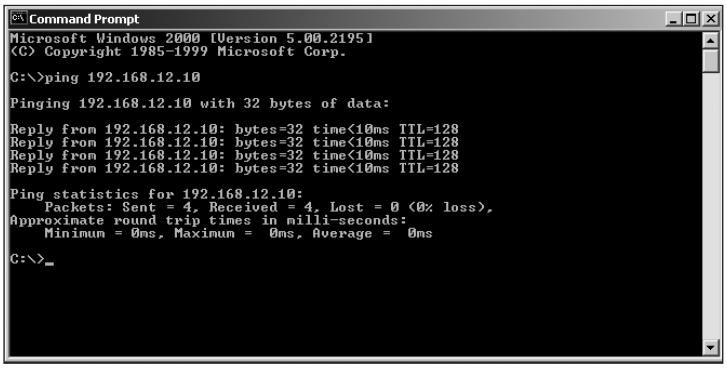



**Figure 2-23** Successful ping replies


## Project 2-4

**To reset win2kpro1 to obtain an IP address dynamically via DHCP:**

1. Right-click **My Network Places** and choose **Properties**.
2. Right-click **Local Area Connection** and choose **Properties**.
3. Click **Internet Protocol (TCP/IP)** and then click the **Properties** button.
4. In the Internet Protocol (TCP/IP) Properties dialog box (shown in Figure 2-21), select **Obtain an IP address automatically**.
5. Click **OK** to close the Internet Protocol (TCP/IP) Properties dialog box.

6. Click **OK** to close the Local Area Connection Properties box.
7. Close the Networking and Dialup Connections dialog box by clicking the **Close** button in its upper-right corner.


## Project 2-5

**To install the NWLink IPX/SPX protocol stack:**

1. Right-click **My Network Places** and choose **Properties**.

   See Figure 2-20.
2. Right-click **Local Area Connection** and choose **Properties**.
3. Click **Install** button to open the Select Network Component Type dialog box shown in Figure 2-24.

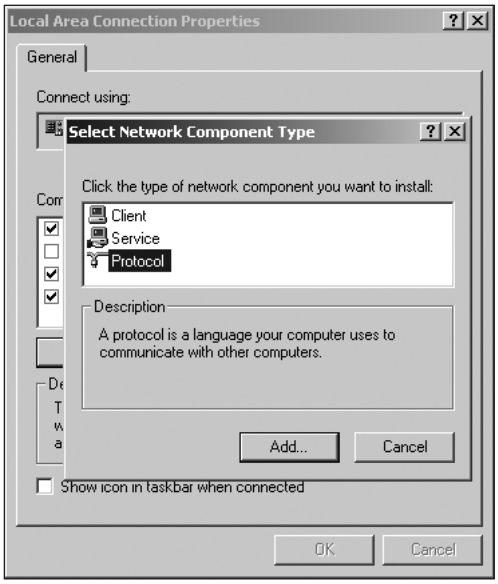



**Figure 2-24** Select Network Component Type dialog box

4. Select **Protocol** and then click **Add**.

   After a short pause, the Select Network Protocol dialog box shown in Figure 2-25 opens.
5. Select **NWLink IPX/SPX/NetBIOS Compatible Transport Protocol**, and then click **OK**.
6. Click **Close** to close the Local Area Connection Properties dialog box.
7. Close Networking and Dialup Connections by clicking the **Close** button in its upper-right corner.

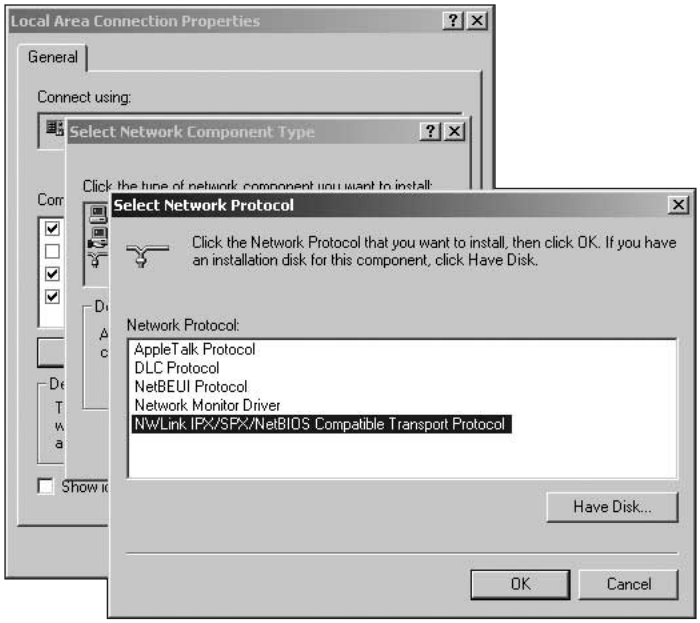



**Figure 2-25** Select Network Protocol dialog box


## Project 2-6

Although TCP/IP is installed by default with the Typical Network settings during a Windows 2000 installation, you must still know how to install the protocol in case someone accidentally removes it.

**To install the TCP/IP protocol stack:**

1. Right-click **My Network Places** and choose **Properties**.

   See Figure 2-20.

2. Right-click **Local Area Connection** and choose **Properties**.
3. Click **Install** to open the Select Network Component Type dialog box shown in Figure 2-24.
4. Select **Protocol** and then click **Add**.

   After a short pause, the Select Network Protocol dialog box shown in Figure 2-25 opens.

5. Select **Internet Protocol (TCP/IP)** and then click **OK**.
6. Click **Close** to close the Local Area Connection Properties dialog box.
7. Close the Networking and Dialup Connections dialog box by clicking the **Close** button in its upper-right corner.

## Project 2-7

**To configure TCP/IP packet filtering to allow only incoming http over TCP:**

1. Right-click **My Network Places** and choose **Properties**.
2. Right-click **Local Area Connection** and choose **Properties**.
3. Click **Internet Protocol (TCP/IP)** and then click the **Properties** button.
4. Click the **Advanced** button.
   You should see the Advanced TCP/IP Settings, Options tab illustrated in Figure 2-26.
5. Click the **Options** tab.

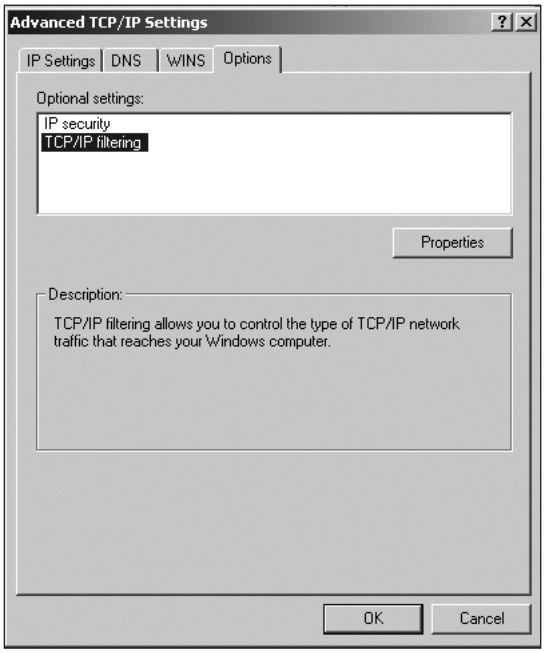



**Figure 2-26** Advanced TCP/IP Settings, Options tab

6. Click **TCP/IP Filtering** to select it, and then click the **Properties** button.
   You should see the TCP/IP Filtering dialog box shown in Figure 2-27.
7. Click the **Enable TCP/IP Filtering (All adapters)** check box.
8. Select the **Permit Only** radio button over **TCP Ports**.
9. Click the **Add** button.
   The Add Filter box shown in Figure 2-28 appears.

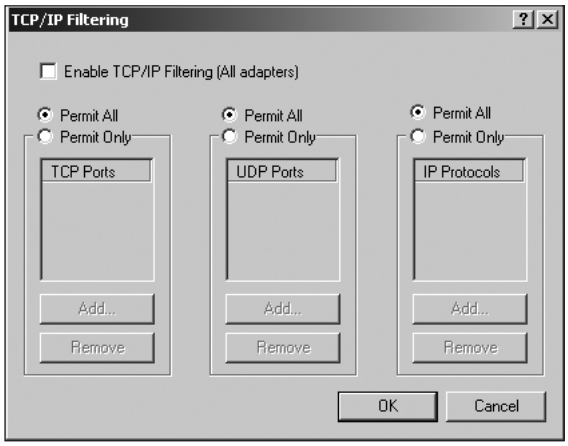



**Figure 2-27** TCP/IP Filtering dialog box

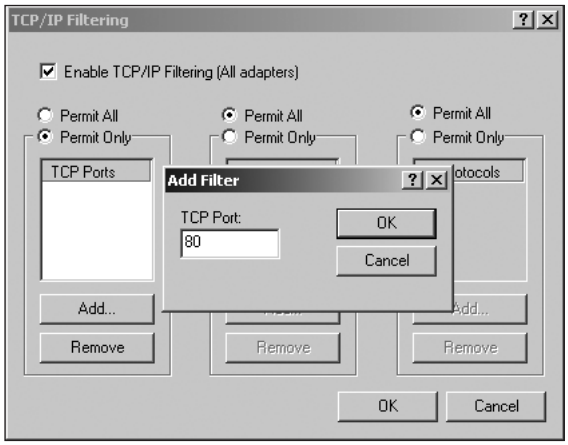



**Figure 2-28** Add Filter entry box

10. Type **80** for the TCP Port.
11. Click **OK** to close the Add Filter dialog box.
12. Click **OK** to close the TCP/IP Filtering dialog box.
13. Click **OK** to close the Advanced TCP/IP Settings dialog box.
14. Click **OK** to close the Internet Protocol (TCP/IP) Properties dialog box.
15. Click **OK** to close the **Local Area Connection Properties** dialog box.

    You see a warning box with the text: "You must shut down and restart your computer before the new settings will take effect. Do you want to restart your computer now?"
16. Click **Yes** to restart your computer.

## Project 2-8

**To change the binding order of multiple protocols:**

1. Right-click **My Network Places** and choose **Properties**.
2. Click the **Advanced** command shown in Figure 2-29.

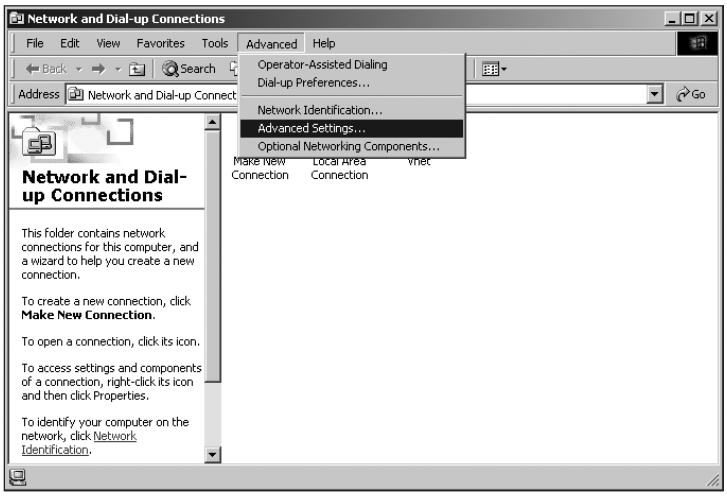



**Figure 2-29** Advanced menu

3. Click **Advanced Settings**.

   This opens the Advanced Settings dialog box with the Adapters and Bindings tab shown in Figure 2-30.
4. Under Client for Microsoft Networks, click **Internet Protocol (TCP/IP)** and then click the rightmost **up arrow** to move TCP/IP up in the binding order.
5. Click **OK** to close the Advanced Settings dialog box.
6. Close the Networking and Dialup Connections dialog box by clicking the **Close** button in its upper-right corner.

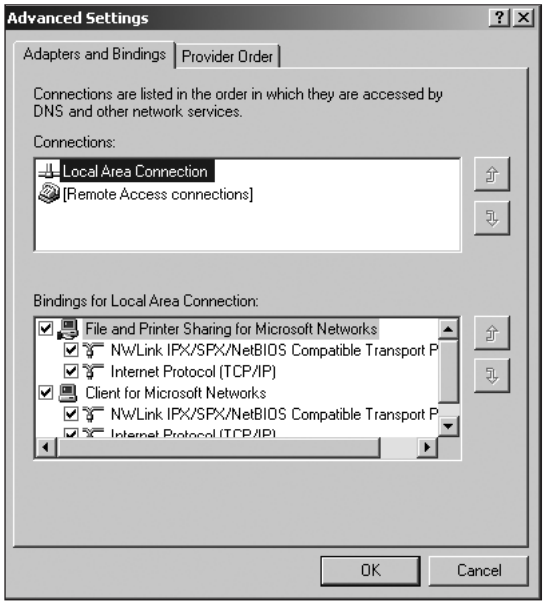



**Figure 2-30** Advanced Settings, Adapters and Bindings tab

## CASE PROJECTS

### Case 1

Your network administrator decides to use the private class B address 172.16.0.0 for a network redesign. The network currently has 30 subnets, and its size is expected to double in the next two years. Using the seven steps of subnetting, design a subnetting scheme that meets the company's present situation while allowing for future growth.

### Case 2

A fairly large corporation hired you to develop a plan for IP address allocation. During a meeting, the CIO states that every machine on the network should use dynamic address assignment. Prepare a short summary about both dynamic and static IP address assignment. In the summary, clarify any problems the CIO's plan may cause.

### Case 3

At Freytech Inc., the IS department needs simple tools to monitor and manage its Windows 2000 machines. Prepare a list of the tools currently available in Windows 2000, and describe the function of each.